Taming Video Diffusion Prior with Scene-Grounding Guidance for 3D Gaussian Splatting from Sparse Inputs

Supplementary Material

The project page is available at HERE, which contains demo videos for better visualization.

A. Implementation Details

During the denoising sampling process, we employ the DDIM sampler [10] combined with our proposed guidance, setting the number of sampling steps to 50. Regarding the trajectory initialization strategy, for each input view in its camera space, we sample views by changing the polar/azimuth angle to $[-30°, -15°, 0°, 15°, 30°]$, and setting the radial distance to $[1, \frac{1}{3}, \frac{1}{10}]$ of the depth of the center pixel (from the prediction of ViewCrafter [17]). Out of 75 sampled views, we discard those whose renderings exhibit holes larger than 10% of the image size (to filter out uncommon viewpoints), then select the top 6 views with the largest holes from the remaining. To obtain the point cloud used for initialization, we follow the standard pipeline provided on the DUSt3R [13] webpage. Since our focus is sparse-input radiance fields reconstruction, the groundtruth camera poses and intrinsics are provided. During DUSt3R optimization, we fix both the poses and intrinsics to their groundtruth values. In the main paper, we conduct experiments on a new benchmark that is created from two indoor datasets, the synthetic Replica [11] and the realistic ScanNet++ [16] datasets. Please refer to [18] for more details about the benchmark.

B. More Results

Our method focuses on holistic modeling of an indoor scene of a moderate size, and we conduct the experiments in the main paper with 6 input views, since 6 input views are basically sufficient to cover the entire room. To validate the effectiveness of our method, we also test our method with different number of views following the common 3/6/9-view settings of sparse-input modeling. Tab. A1 validates that, our method is effective given different number of input views, with consistent improvements over our baseline. InstantSplat [2] is a strong baseline of sparse-input pose-free modeling, leveraging DUSt3R [13] point cloud for 3DGS initialization. Our method also consistently outperforms InstantSplat as shown in Tab. A1.

To obtain a thorough understanding of the source of the performance improvement, we show some quantitative results regarding performances of observable and the unobservable regions respectively in Tab. A2. The results show that our method brings improvement in both observable and unobservable regions.

We further compare our method with two representative methods that leverage diffusion models for sparse-input modeling, ReconFusion [14] and CAT3D [3] on the datasets of RealEstate10K and LLFF. We adhere to their settings for fair comparisons and the results are shown in Tab. A3. On the LLFF dataset, our method is based on the strong baseline of binocular-guided 3DGS [4]. The results show that our method achieves comparable performance with both ReconFusion and CAT3D.

We provide per-scene comparisons in Table A4, demonstrating that our method consistently achieves superior performance across all scenes. Additional qualitative results are shown in Fig. A3. These results highlight the effectiveness of our approach in addressing issues such as extrapolation and occlusion, as seen in examples like the wall behind the chair (second row) and the ceiling (third row). Furthermore, our method preserves more intact structures with finer details, such as the edges in the fifth and sixth rows.

We present a comparison of the generated sequences from the video diffusion model with and without the proposed guidance in Fig. A2. The results clearly show that our proposed guidance enhances the plausibility of the generated sequences by maintaining consistent appearances and ensuring that only elements present in the scene are generated. Consistency in the generated video is crucial for effective 3DGS optimization. Using inconsistent sequences for 3DGS optimization often leads to artifacts, such as black shadows in the renderings, which significantly degrade visual quality, as demonstrated on the demo page.

C. Discussion

While our approach significantly improves overall quality by addressing extrapolation and occlusion challenges, we observe that it occasionally produces over-smoothed results. We hypothesize that this is due to the limited resolution supported by the video diffusion model during generation. On a 32GB V100 GPU, we are constrained to generating sequences at resolutions of 320×448 for the Replica dataset and 320×512 for the ScanNet++ dataset, which are subsequently upsampled to rendering resolutions of 480×640 and 480×720, respectively, for supervision during 3DGS optimization. This upsampling process introduces undersampling, which can smooth out certain regions and result in over-smoothed effects. Addressing the challenge of preserving high-frequency details during 3DGS optimization under resource-limited sequence generation remains an open problem and is a direction for future work.

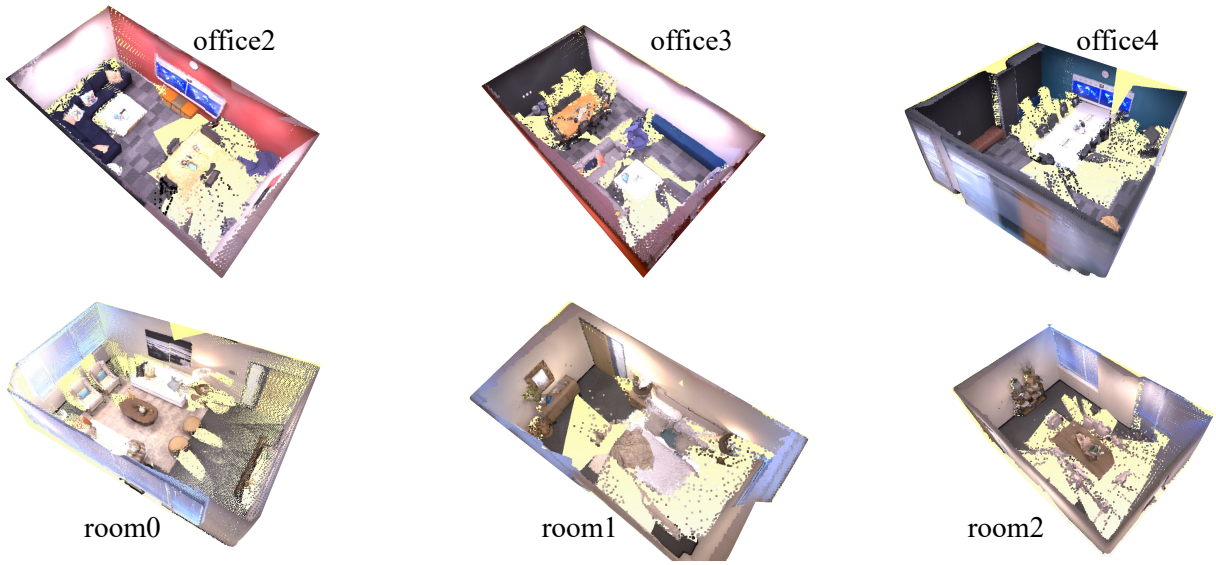

Figure A1. Point clouds from DUSt3R [13] optimized with sparse input views on the Replica dataset. The yellow parts represent unobserved regions, e.g., regions that are outside the field of view or occluded. Note that the ceilings are removed for better visualization.

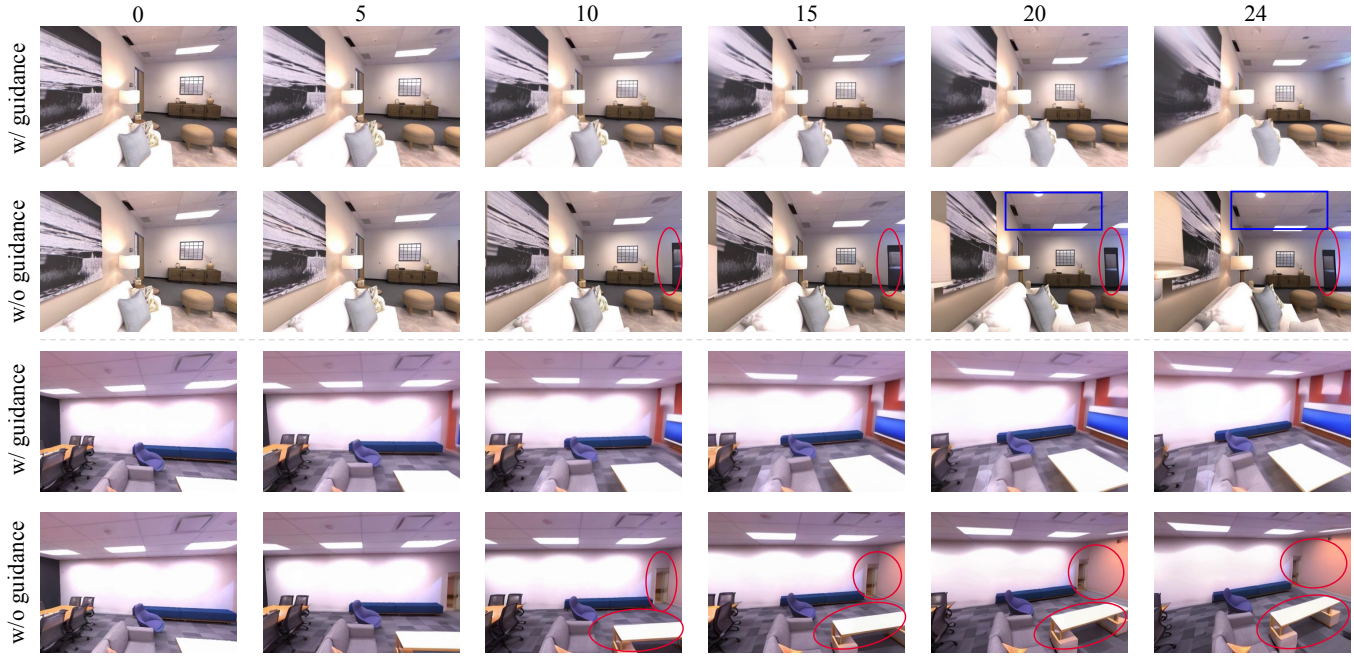

Figure A2. Generated frames from the video diffusion model with and without the proposed guidance. The numbers at the top indicate the frame IDs. The first frame corresponds to an image from the sparse input views, while other frames are generated. Without guidance, the generated sequences exhibit significant inconsistencies: **(i)** appearance inconsistencies, highlighted by the blue boxes; and **(ii)** hallucinated elements that do not exist in the scene, highlighted by the red boxes. In contrast, with the proposed guidance, the generated sequences are more plausible and consistent.

	3-view			6-view			9-view		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
Replica									
Baseline 3DGS	19.87	0.794	0.178	22.80	0.818	0.179	24.81	0.863	0.124
InstantSplat [2]	20.49	0.766	0.226	20.35	0.760	0.290	18.44	0.708	0.373
Ours	23.98	0.848	0.136	26.35	0.872	0.122	27.42	0.891	0.111

Table A1. Our method brings performance improvement over the baseline with different number of input views, and consistently outperforms another strong sparse-input modeling baseline InstantSplat [2].

Replica 6-view	Full Image			Observable Regions			Unobservable Regions		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
Baseline 3DGS	22.80	0.818	0.179	25.45	0.860	0.129	14.27	0.967	0.029
w/ Vanilla Generation	23.69	0.840	0.160	25.00	0.870	0.119	17.11	0.977	0.025
Ours	26.35	0.872	0.122	27.12	0.894	0.091	20.85	0.985	0.020
Baseline 3DGS+LaMa [12]	24.56	0.833	0.167	25.45	0.860	0.129	17.80	0.981	0.021
Baseline 3DGS+SDInpaint [9]*	25.15	0.853	0.141	26.13	0.878	0.104	19.25	0.982	0.022

Table A2. Analysis of performance regarding observable and unobservable regions. * refers to incorporating our trajectory initialization strategy. The methods in the second block utilize inpainting models.

	3-view			6-view			9-view		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
RealEstate10K									
ReconFusion [14]	25.84	0.910	0.144	29.99	0.951	0.103	31.82	0.961	0.092
CAT3D [3]	26.78	0.917	0.132	31.07	0.954	0.092	32.20	0.963	0.082
Ours	25.03	0.871	0.136	30.62	0.944	0.069	32.45	0.955	0.062
LLFF									
ReconFusion [14]	21.34	0.724	0.203	24.25	0.815	0.152	25.21	0.848	0.134
CAT3D [3]	21.58	0.731	0.181	24.71	0.833	0.121	25.63	0.860	0.107
Ours	21.35	0.746	0.173	25.13	0.851	0.102	26.29	0.880	0.084

Table A3. Comparisons with ReconFusion [14] and CAT3D [3] on the RealEstate10K and LLFF datasets.

	ScanNet++ [16]					Replica [11]						
	a2ccc	*8a20d*	*94ee1*	*78318*	*avg*	*office2*	*office3*	*office4*	*room0*	*room1*	*room2*	*avg*
Mip-NeRF [1]	18.28 0.759 0.351	23.48 0.799 0.321	16.93 0.725 0.431	19.63 0.735 0.451	19.58 0.755 0.389	17.43 0.539 0.486	19.04 0.685 0.421	19.08 0.727 0.393	17.46 0.762 0.342	16.57 0.721 0.386	19.16 0.808 0.317	18.12 0.707 0.391
InfoNeRF [7]	13.90 0.662 0.468	17.69 0.691 0.437	14.34 0.627 0.516	12.21 0.605 0.558	14.54 0.646 0.495	13.66 0.463 0.612	12.53 0.545 0.623	11.51 0.592 0.624	12.58 0.618 0.542	14.11 0.689 0.435	14.00 0.678 0.477	13.07 0.598 0.552
DietNeRF [5]	20.67 0.751 0.385	23.00 0.776 0.363	15.34 0.627 0.516	20.02 0.725 0.459	19.76 0.719 0.431	19.12 0.612 0.458	19.35 0.695 0.417	18.97 0.419 0.721	19.84 0.783 0.34	17.18 0.749 0.386	19.46 0.797 0.343	18.99 0.676 0.444
FreeNeRF [15]	19.93 0.759 0.307	22.37 0.791 0.299	19.42 0.762 0.417	18.94 0.711 0.449	20.17 0.756 0.368	20.89 0.688 0.359	21.06 0.735 0.340	20.25 0.750 0.364	22.55 0.831 0.234	19.69 0.781 0.325	21.43 0.807 0.321	20.99 0.765 0.324
S^3NeRF [18]	21.81 0.801 0.324	25.60 0.811 0.330	20.05 0.784 0.357	21.36 0.753 0.444	22.21 0.787 0.364	22.79 0.728 0.326	23.83 0.773 0.309	23.08 0.801 0.301	24.01 0.862 0.213	19.66 0.808 0.277	21.87 0.825 0.293	22.54 0.800 0.287
3DGS$^{\updownarrow}$ [6]	20.65 0.824 0.193	23.49 0.857 0.136	20.38 0.821 0.218	21.11 0.764 0.298	21.41 0.817 0.211	25.03 0.873 0.141	23.60 0.858 0.147	22.14 0.834 0.180	20.32 0.720 0.204	22.68 0.802 0.203	23.07 0.824 0.196	22.80 0.818 0.179
DNGaussian [8]	19.10 0.765 0.343	21.21 0.781 0.292	17.55 0.743 0.382	18.20 0.730 0.450	19.01 0.755 0.367	22.68 0.843 0.233	18.40 0.789 0.291	12.31 0.644 0.628	12.60 0.534 0.722	18.87 0.708 0.397	20.91 0.790 0.338	17.63 0.718 0.435
DNGaussian$^{\updownarrow}$ [8]	20.47 0.805 0.213	23.73 0.842 0.183	18.90 0.784 0.287	19.61 0.722 0.357	20.68 0.788 0.281	25.31 0.890 0.124	23.34 0.853 0.161	21.83 0.837 0.197	20.33 0.729 0.226	22.59 0.800 0.208	22.88 0.820 0.219	22.71 0.821 0.189
FSGS [19]	19.19 0.760 0.321	18.98 0.735 0.316	15.77 0.719 0.415	17.87 0.708 0.442	17.95 0.730 0.373	20.70 0.802 0.266	20.26 0.790 0.255	21.62 0.825 0.271	19.65 0.654 0.315	19.23 0.712 0.374	19.89 0.779 0.342	20.22 0.760 0.304
FSGS$^{\updownarrow}$ [19]	21.28 0.826 0.219	22.56 0.844 0.193	20.28 0.815 0.267	20.79 0.767 0.350	21.23 0.813 0.257	24.37 0.873 0.194	23.41 0.856 0.174	23.45 0.862 0.189	21.02 0.759 0.198	23.56 0.823 0.205	22.14 0.822 0.270	22.99 0.833 0.205
Ours	**25.21** **0.857** **0.157**	**25.10** **0.882** **0.118**	**23.10** **0.860** **0.201**	**22.16** **0.803** **0.269**	**23.89** **0.850** **0.182**	**27.46** **0.916** **0.083**	**26.81** **0.902** **0.099**	**27.43** **0.897** **0.122**	**24.85** **0.796** **0.145**	**26.00** **0.851** **0.142**	**25.53** **0.872** **0.142**	**26.35** **0.872** **0.122**

Table A4. Per-scene performance of various models on the ScanNet++ and Replica datasets. For each method, the three rows represent PSNR, SSIM, and LPIPS, respectively. *avg* indicates the average performance across all scenes in each dataset. Including our approach, 3DGS-based methods marked with $^{\updownarrow}$ are initialized with the point cloud from DUSt3R [13].

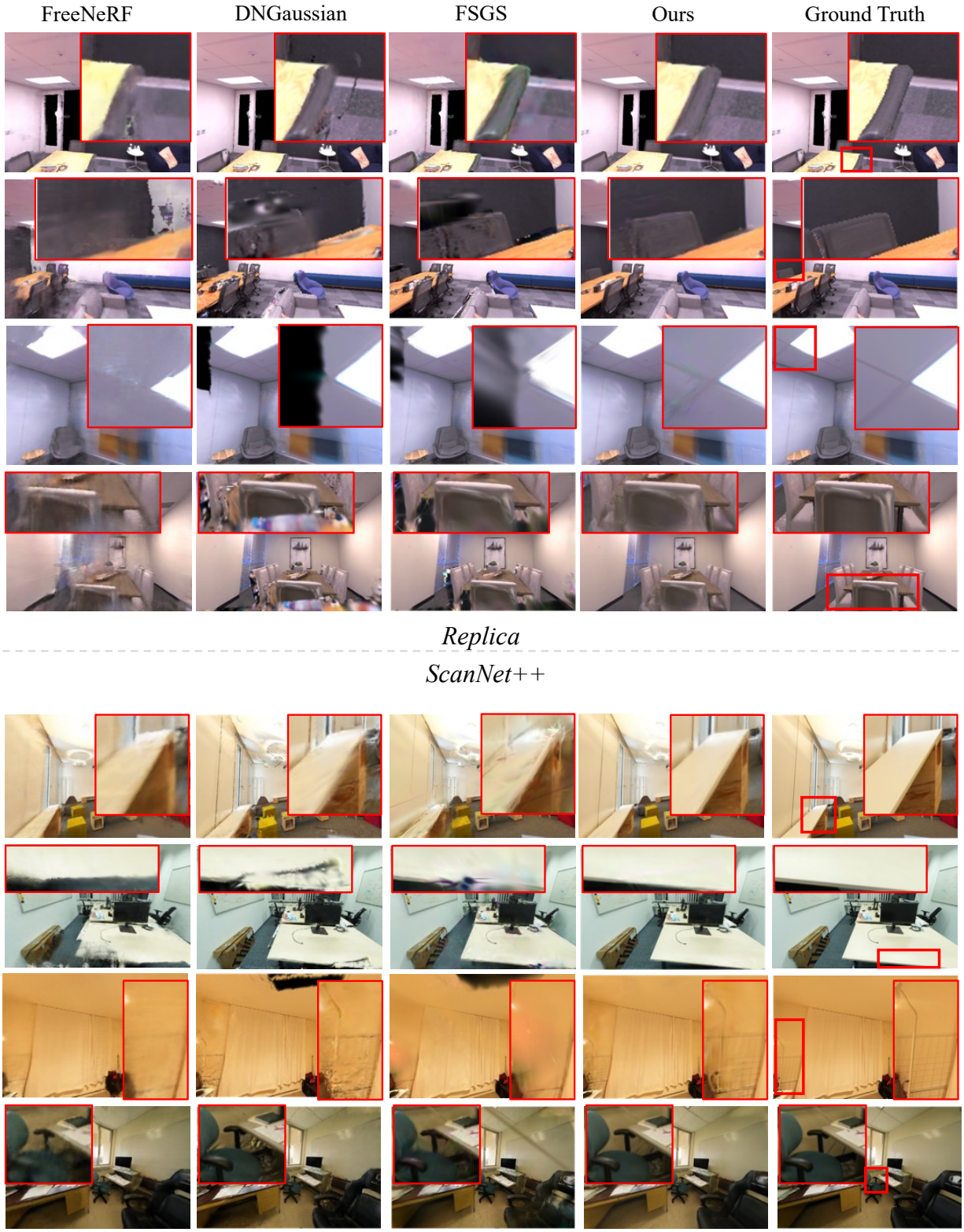

Figure A3. Qualitative comparisons between other works on Replica and ScanNet++ datasets. All 3DGS-based methods are optimized using the initialized point cloud from DUSt3R [13].

References

[1] Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *ICCV*, 2021. 3

[2] Zhiwen Fan, Wenyan Cong, Kairun Wen, Kevin Wang, Jian Zhang, Xinghao Ding, Danfei Xu, Boris Ivanovic, Marco Pavone, Georgios Pavlakos, et al. Instantsplat: Unbounded sparse-view pose-free gaussian splatting in 40 seconds. *arXiv preprint arXiv:2403.20309*, 2024. 1, 2

[3] Ruiqi Gao, Aleksander Holynski, Philipp Henzler, Arthur Brussee, Ricardo Martin-Brualla, Pratul Srinivasan, Jonathan T Barron, and Ben Poole. Cat3d: Create anything in 3d with multi-view diffusion models. In *NeurIPS*, 2024. 1, 3

[4] Liang Han, Junsheng Zhou, Yu-Shen Liu, and Zhizhong Han. Binocular-guided 3d gaussian splatting with view consistency for sparse view synthesis. In *NeurIPS*, 2024. 1

[5] Ajay Jain, Matthew Tancik, and Pieter Abbeel. Putting nerf on a diet: Semantically consistent few-shot view synthesis. In *ICCV*, 2021. 3

[6] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023. 3

[7] Mijeong Kim, Seonguk Seo, and Bohyung Han. Infonerf: Ray entropy minimization for few-shot neural volume rendering. In *CVPR*, 2022. 3

[8] Jiahe Li, Jiawei Zhang, Xiao Bai, Jin Zheng, Xin Ning, Jun Zhou, and Lin Gu. Dngaussian: Optimizing sparse-view 3d gaussian radiance fields with global-local depth normalization. In *CVPR*, 2024. 3

[9] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. 3

[10] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 1

[11] Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, et al. The replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019. 1, 3

[12] Roman Suvorov, Elizaveta Logacheva, Anton Mashikhin, Anastasia Remizova, Arsenii Ashukha, Aleksei Silvestrov, Naejin Kong, Harshith Goka, Kiwoong Park, and Victor Lempitsky. Resolution-robust large mask inpainting with fourier convolutions. In *WACV*, 2022. 3

[13] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *CVPR*, 2024. 1, 2, 3, 4

[14] Rundi Wu, Ben Mildenhall, Philipp Henzler, Keunhong Park, Ruiqi Gao, Daniel Watson, Pratul P Srinivasan, Dor Verbin, Jonathan T Barron, Ben Poole, et al. Reconfusion: 3d reconstruction with diffusion priors. In *CVPR*, 2024. 1, 3

[15] Jiawei Yang, Marco Pavone, and Yue Wang. Freenerf: Improving few-shot neural rendering with free frequency regularization. In *CVPR*, 2023. 3

[16] Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. Scannet++: A high-fidelity dataset of 3d indoor scenes. In *ICCV*, 2023. 1, 3

[17] Wangbo Yu, Jinbo Xing, Li Yuan, Wenbo Hu, Xiaoyu Li, Zhipeng Huang, Xiangjun Gao, Tien-Tsin Wong, Ying Shan, and Yonghong Tian. Viewcrafter: Taming video diffusion models for high-fidelity novel view synthesis. *arXiv preprint arXiv:2409.02048*, 2024. 1

[18] Yingji Zhong, Kaichen Zhou, Zhihao Li, Lanqing Hong, Zhenguo Li, and Dan Xu. Empowering sparse-input neural radiance fields with dual-level semantic guidance from dense novel views. *arXiv preprint arXiv:2503.02230*, 2025. 1, 3

[19] Zehao Zhu, Zhiwen Fan, Yifan Jiang, and Zhangyang Wang. Fsgs: Real-time few-shot view synthesis using gaussian splatting. In *ECCV*, 2024. 3