Supplementary Materials
LighthouseGS: Indoor Structure-aware 3D Gaussian Splatting for Panorama-Style Mobile Captures

Seungoh Han[1*] Jaehoon Jang[1*] Hyunsu Kim[2] Jaeheung Surh[2]
Junhyung Kwak[2] Hyowon Ha[2†] Kyungdon Joo[1†]
[1]Ulsan National Institute of Science and Technology (UNIST) [2]Bucketplace
{seungohhan00, wkdwogns1997, gustnxodjs, jaeheungsurh
junhk0914, hyowonha.phd, kdjoo369}@gmail.com

Overview

This supplementary material contains additional information not included in the main paper due to space limitations.

- We conduct additional ablation studies on stable pruning, camera pose refinement, and computational efficiency in Sec. 1.
- We demonstrate that `LighthouseGS` generalizes to the standard benchmark and non-planar structures, showing its robustness across diverse scenarios in Sec. 2.
- We provide the collected dataset examples and additional qualitative/quantitative comparisons in Sec. 3.
- We present panorama view synthesis results along with other examples of object placement in Sec. 4.
- We clarify the limitations and future works of the proposed methods in Sec. 5.
- More results are available in the supplementary video.

1. Additional Ablation Study

1.1. Stable Pruning

As shown in Fig. 5 of the main paper, our stable pruning strategy retains reliable 3D Gaussians based on their opacity values during optimization. To evaluate the effect of this criterion, we perform an ablation study by varying the opacity threshold t (see Table 1). Empirically, we set the threshold value to 0.5, which yields the best performance across metrics. Despite a minor increase in LPIPS, the performance gains in PSNR and SSIM are more substantial, resulting in enhanced overall rendering quality. In addition, stable optimization affects preserving accurate scene geometry while removing artifacts in textureless regions (see Fig.7 in the main paper). Even though the numerical impact of this module appears minor, it plays a crucial role in applications like panoramic view synthesis (see Fig. 3). In addition, we conduct an ablation study for each module on all scenes in real and synthetic datasets, as shown in Table 6.

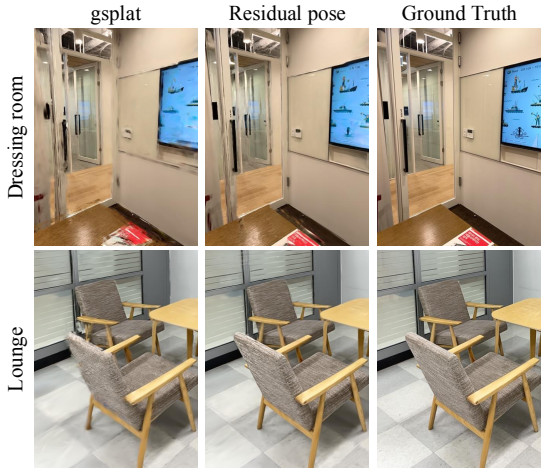

Figure 1. Effect of pose refinement.

Methods	PSNR ↑	SSIM ↑	LPIPS ↓
w/o stable pruning	24.25	0.799	**0.313**
ours ($t = 0.9$)	<u>24.77</u>	0.805	0.317
ours ($t = 0.7$)	24.68	<u>0.807</u>	<u>0.316</u>
ours ($t = 0.5$)	**25.06**	**0.809**	0.317

Table 1. Ablation on stable pruning. All reported metrics represent the average performance across all real scenes.

1.2. Camera Pose Refinement

To validate the effectiveness of residual pose refinement, we compare three configurations: (1) no pose optimiza-

*Equal contribution
†Corresponding author

Scene	w/o refinement			Direct pose refinement			Residual pose refinement		
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
Meeting room	20.62	0.604	0.445	<u>21.73</u>	<u>0.692</u>	<u>0.420</u>	**23.56**	**0.721**	**0.405**
Dressing room	18.21	0.661	0.431	<u>20.24</u>	<u>0.712</u>	<u>0.388</u>	**21.93**	**0.738**	**0.346**
Pantry	23.25	0.850	0.339	<u>26.10</u>	**0.888**	<u>0.291</u>	**26.80**	**0.888**	**0.287**
Lounge	20.60	0.707	0.346	<u>24.60</u>	<u>0.805</u>	<u>0.301</u>	**25.00**	**0.807**	**0.295**
Conference room	23.72	0.840	0.311	<u>27.42</u>	<u>0.890</u>	<u>0.253</u>	**28.01**	**0.892**	**0.252**

Table 2. Pose refinement ablations on the real-world dataset. Compared to other approaches, our residual pose refinement achieves superior performance across all metrics. We highlight the best scores as bold.

tion (w/o refinement), (2) direct pose optimization via gsplat [10], and (3) our residual pose refinement. gsplat computes the gradients of input camera poses via differentiable rasterization, which enables camera poses to be updated to enhance rendering quality. However, directly optimizing camera poses with gsplat often leads to suboptimal results during the update process, resulting in blurry artifacts in rendered images (see Fig. 1). In contrast, our proposed residual pose refinement outperforms both the baseline (*i.e.*, w/o refinement) and direct pose optimization with gsplat by updating only the residuals of camera poses. We also present quantitative comparisons across all metrics, as shown in Table 2. As a result, residual pose refinement further improves rendering performance both qualitatively and quantitatively.

1.3. Computational Efficiency

To evaluate the efficiency of the proposed method, we measure training and inference speed as well as memory usage (see Table 3). While DNGaussian [6] offers the fastest training and inference speeds with low memory usage, it suffers from lower rendering quality. Scaffold-GS [8] and GeoGaussian [7] show comparable rendering performance with DNGaussian, but they require significantly increased training time. On the other hand, our method achieves significantly higher rendering quality while maintaining competitive training time, inference speed, and memory usage.

Methods	PSNR ↑	Train ↓	FPS ↑	Mem. ↓
Baseline	16.55	30m	163	608M
DNGaussian	17.74	**15m**	**290**	**215M**
ScaffoldGS	<u>17.85</u>	112m	<u>224</u>	256M
GeoGaussian	17.55	134m	138	1.3G
Ours	**21.93**	<u>19m</u>	204	<u>224M</u>

Table 3. Ablation on computational efficiency. We mark the best and second scores as bold and underline, respectively.

Methods	PSNR ↑	SSIM ↑	LPIPS ↓
3DGS	17.96	0.601	0.451
Ours	**20.33**	**0.664**	**0.377**

Table 4. Ablation on few-shot ScanNet.

2. Generalization

2.1. Generalization on ScanNet

We further evaluate the robustness and generalization ability of `LighthouseGS` on the ScanNet [2] benchmark under a few-shot setting. This setting is also practical in real-world applications, where only a sparse set of training views is available. Specifically, we utilize *scene0710*, with 18 views for training and 8 views for evaluation. Compared to panorama-style motion, the few-shot setting involves wide baselines and limited overlap between frames, making view interpolation significantly challenging. Nevertheless, as shown in Table 4, our method achieves more robust performance than the baseline 3DGS.

2.2. Generalization to Non-Planar Structures

While `LighthouseGS` leverages planar structures throughout the pipeline, it generalizes well to non-planar environments. Before plane-wise local alignment, we perform image-wise global alignment that does not rely on planar structures, ensuring global point alignment even in clutter or curved surfaces. Moreover, since the photometric loss dominates the objective function, Gaussians not initialized from plane scaffolds can produce high-fidelity renderings. As shown in Fig.7 of the main paper, our method successfully reconstructs scenes with various non-planar surfaces such as soft furnishings (e.g., curtains) and curved objects (e.g., vase), demonstrating generalization beyond planar structures.

3. Extended Evaluations and Data Examples

3.1. Data Examples

As existing indoor scene datasets [1, 2, 9] do not support panorama-style motion, we construct a new dataset in both real and synthetic scenes to evaluate `LighthouseGS`.

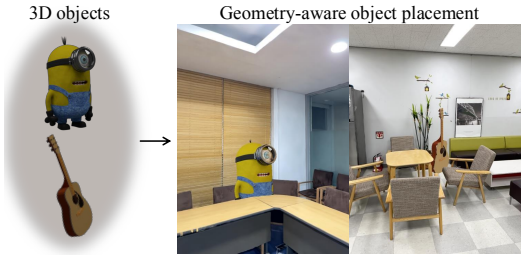

Figure 2. Object Placement. Given 3D virtual objects, it can be naturally inserted into the desired location in the optimized scene of LighthouseGS.

Figure 3. Comparison on panoramic view synthesis. Our implemented baseline struggles with severe artifacts and color inconsistencies. On the other hand, LighthouseGS produces visually consistent panoramic views with fewer geometric artifacts and improved color consistency.

The real-world dataset includes five indoor environments: Veranda, Dressing Room, Pantry, Cafeteria, and Meeting Room. Similarly, the synthetic dataset contains five virtual scenes: Playroom, Bedroom, Primary Bedroom, Living Room, and Studio. We visualize a panorama-style camera motion used to capture the scene alongside sample images (see Fig. 4).

3.2. Qualitative Results

Figure 6 and Figure 7 present additional qualitative evaluations on the real-world and synthetic datasets, respectively. Overall, LighthouseGS achieves photorealistic novel view synthesis while better preserving accurate scene geometry compared to other methods. In particular, our method effectively removes blurry artifacts caused by motion drift and eliminates floaters in non-textured regions by leveraging planar structures throughout the entire pipeline.

3.3. Quantitative Results

We provide additional quantitative evaluations for each scene in the real-world and synthetic datasets (see Table 5). As a result, LighthouseGS outperforms previous approaches across all metrics.

4. Applications

4.1. Object Placement

In addition to an example in the main paper, we provide additional object placement results using other objects and scenes (see Fig. 2). Accurate scene geometry is essential for aligning virtual objects seamlessly with real-world environments. Leveraging the planar structure of indoor scenes, LighthouseGS enables not only high-fidelity rendering but also precise geometric reconstruction, such as depth and normals. Furthermore, directly placing virtual assets into the 3DGS scene causes color inconsistencies. To resolve this, we render a novel panoramic view from LighthouseGS and synthesize it into an HDR map for photorealistic relighting. Consequently, by utilizing these accurate depth maps as alpha values and the HDR map, we can naturally place virtual objects at desired locations.

4.2. Panoramic View Synthesis

We visualize panoramic view synthesis from an unseen viewpoint, as shown in Fig. 5. Beyond the inherent constraints on mobile capture and panorama-style motion, LighthouseGS better represents the surrounding panorama compared to the baseline (see Fig. 3).

5. Limitations & Future Works

LighthouseGS fundamentally inherits the weakness of 3DGS representations [5]. Specifically, 3DGS-based approaches fail to extrapolate novel views far from the capturing trajectory. Inspired by previous works [3, 4], we believe that this limitation could be addressed by incorporating generative priors. Because our methods are tailored with indoor scene environments, such as utilizing geometric planes, LighthouseGS does not align with open spaces like outdoor environments and driving scenes.

Method	Metric	Real-world Scenes					Synthetic Scenes				
		Meeting	Dressing	Pantry	Lounge	Conf.	Playroom	Bedroom	Prim. Bed	Living	Studio
3DGS†	PSNR	18.99	16.55	22.12	21.80	23.52	24.81	25.85	24.35	20.29	25.18
	SSIM	0.609	0.528	0.803	0.692	0.789	0.866	0.798	0.782	0.740	0.764
	LPIPS	0.486	0.457	0.335	0.358	0.300	0.182	0.248	0.274	0.252	0.302
DNGaussian†	PSNR	19.38	17.74	22.55	22.35	23.49	23.93	25.96	24.14	20.46	24.50
	SSIM	0.641	0.622	0.830	0.713	0.813	0.873	0.822	0.781	0.770	0.784
	LPIPS	0.497	0.467	0.350	0.352	0.306	0.223	0.312	0.349	0.292	0.386
Scaffold-GS†	PSNR	19.76	17.85	22.76	22.10	23.06	24.38	25.07	24.60	19.94	24.20
	SSIM	0.623	0.598	0.818	0.697	0.797	0.852	0.784	0.777	0.729	0.761
	LPIPS	0.442	0.406	0.291	0.296	0.257	0.174	0.260	0.263	0.244	0.315
GeoGaussian†	PSNR	18.53	17.55	22.11	21.66	22.80	24.26	25.88	24.78	20.71	25.05
	SSIM	0.587	0.562	0.811	0.691	0.785	0.854	0.813	0.792	0.759	0.784
	LPIPS	**0.343**	0.372	**0.258**	0.323	0.257	0.204	0.265	0.288	0.256	0.301
Ours	PSNR	**23.56**	**21.93**	**26.80**	**25.00**	**28.01**	**27.74**	**31.64**	**29.48**	**24.67**	**30.79**
	SSIM	**0.721**	**0.738**	**0.888**	**0.807**	**0.892**	**0.919**	**0.902**	**0.893**	**0.890**	**0.885**
	LPIPS	0.405	**0.346**	0.287	**0.295**	**0.252**	**0.107**	**0.102**	**0.131**	**0.117**	**0.153**

Table 5. Quantitative comparisons on real and synthetic datasets. † indicates methods that initialize with plane scaffold assembly. We highlight the best and second performances in **bold** and underline, respectively.

Module	Metric	Real-world Scenes					Synthetic Scenes				
		Meeting	Dressing	Pantry	Lounge	Conf.	Playroom	Bedroom	Prim. Bed	Living	Studio
w/o res. pose	PSNR	20.62	18.21	23.25	20.60	23.72	23.31	25.16	23.55	18.59	24.49
	SSIM	0.604	0.661	0.850	0.707	0.840	0.822	0.765	0.730	0.705	0.744
	LPIPS	0.445	0.431	0.339	0.346	0.311	0.199	0.251	0.280	0.282	0.312
w/o color corr.	PSNR	20.98	20.79	24.64	24.99	26.25	25.13	29.11	27.70	23.35	26.95
	SSIM	0.698	0.733	0.884	**0.807**	0.890	0.877	0.882	0.875	0.857	0.840
	LPIPS	0.439	0.360	0.298	0.297	0.259	0.195	0.150	0.170	0.166	0.204
w/o stable optim.	PSNR	**23.79**	21.81	26.47	22.26	27.51	26.68	31.43	29.41	23.71	30.33
	SSIM	0.716	0.733	0.884	0.777	0.888	0.896	0.901	0.892	0.877	0.881
	LPIPS	**0.399**	0.348	**0.284**	0.311	0.259	0.136	0.103	**0.129**	0.133	**0.147**
Ours	PSNR	23.56	**21.93**	**26.80**	**25.00**	**28.01**	**27.74**	**31.64**	**29.48**	**24.67**	**30.79**
	SSIM	**0.721**	**0.738**	**0.888**	**0.807**	**0.892**	**0.919**	**0.902**	**0.893**	**0.890**	**0.885**
	LPIPS	0.405	**0.346**	0.287	**0.295**	**0.252**	**0.107**	**0.102**	0.131	**0.117**	0.153

Table 6. Ablation on modules. We highlight the best and second performances in **bold** and underline, respectively.

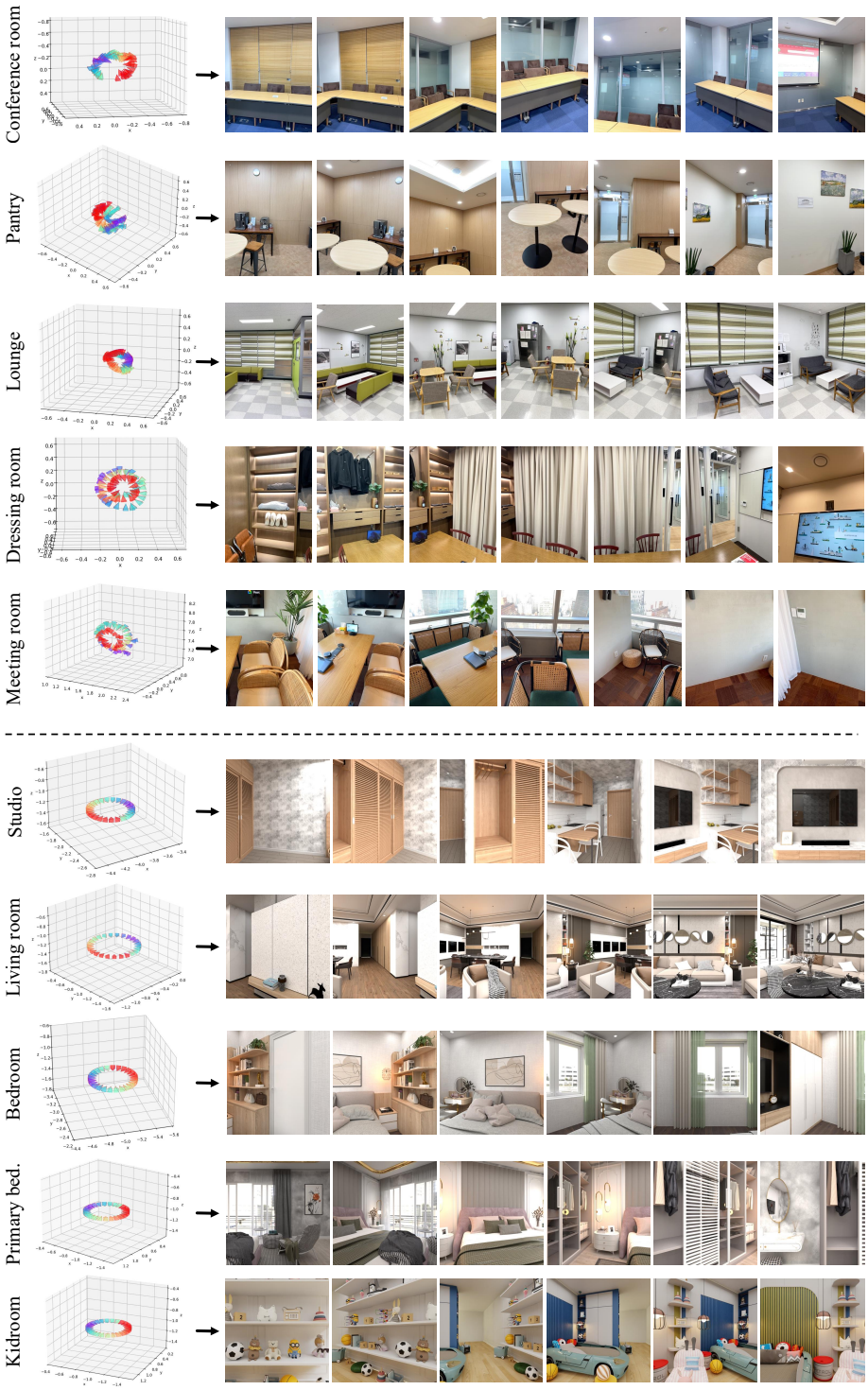

Figure 4. Examples of collected real and synthetic datasets. We visualize sampled images with camera trajectories for each scene. Camera frustums are colored from red (earlier frames) to purple (later frames).

Figure 5. Result of panoramic view synthesis. Thanks to the use of practical panorama-style captures, we achieve panorama-view synthesis from an unseen viewpoint, utilizing all real and synthetic scenes.

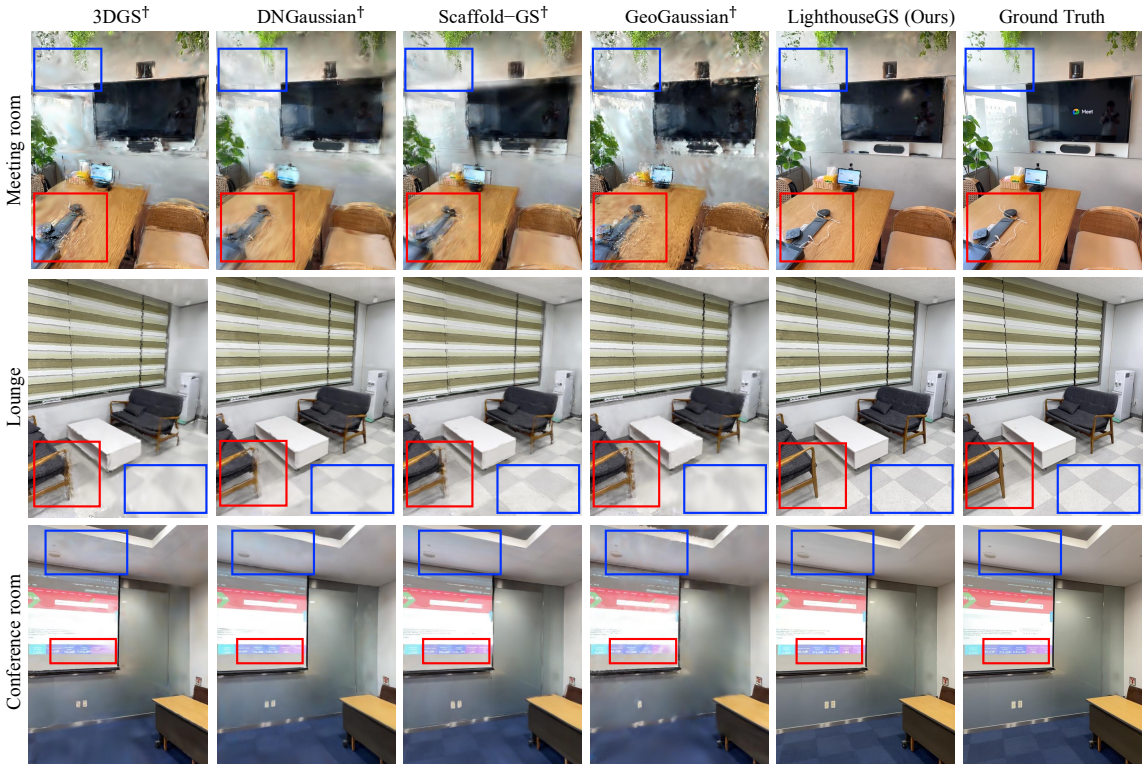

Figure 6. Additional qualitative comparisons on the real-world dataset. The red boxes clearly show the details of textured shapes without blurry artifacts, while the blue boxes represent accurate scene geometry in textureless regions.

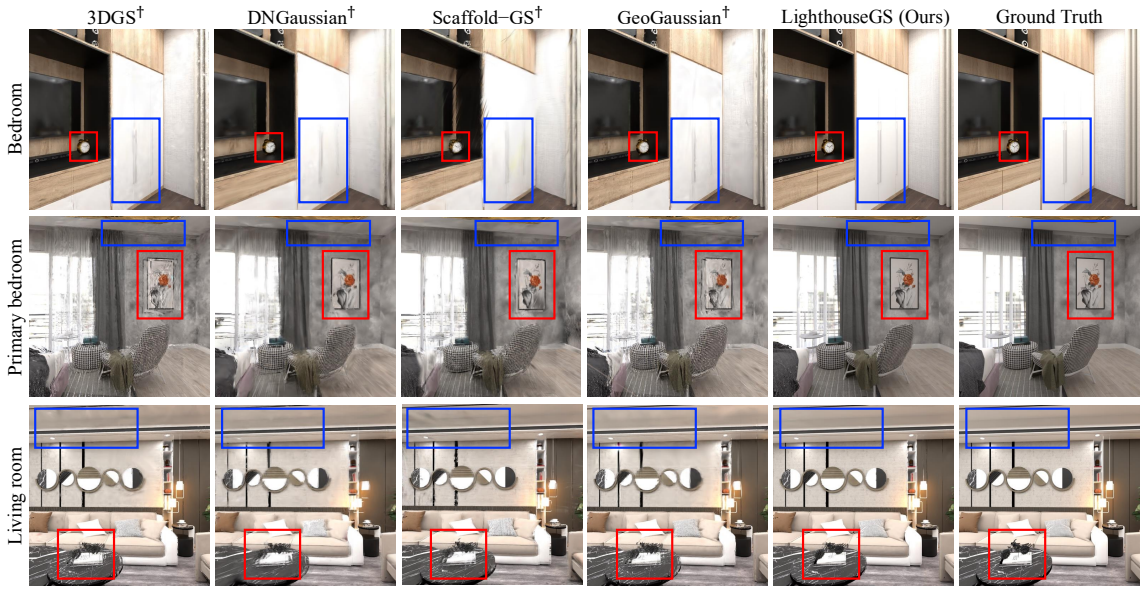

Figure 7. Additional qualitative comparisons on the synthetic dataset. Unlike other methods that produce geometric and visual artifacts, `LighthouseGS` achieves photorealistic rendering without such artifacts.

References

[1] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgb-d data in indoor environments. *arXiv preprint arXiv:1709.06158*, 2017. 2

[2] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *CVPR*, 2017. 2

[3] Hyojun Go, Byeongjun Park, Jiho Jang, Jin-Young Kim, Soonwoo Kwon, and Changick Kim. Splatflow: Multi-view rectified flow model for 3d gaussian splatting synthesis. In *CVPR*, pages 21524–21536, 2025. 3

[4] Hyojun Go, Byeongjun Park, Hyelin Nam, Byung-Hoon Kim, Hyungjin Chung, and Changick Kim. Videorfsplat: Direct scene-level text-to-3d gaussian splatting generation with flexible pose and multi-view joint modeling. *arXiv preprint arXiv:2503.15855*, 2025. 3

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

[6] 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. 2

[7] Yanyan Li, Chenyu Lyu, Yan Di, Guangyao Zhai, Gim Hee Lee, and Federico Tombari. Geogaussian: Geometry-aware gaussian splatting for scene rendering. In *ECCV*, 2024. 2

[8] Tao Lu, Mulin Yu, Linning Xu, Yuanbo Xiangli, Limin Wang, Dahua Lin, and Bo Dai. Scaffold-gs: Structured 3d gaussians for view-adaptive rendering. In *CVPR*, 2024. 2

[9] 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. 2

[10] Vickie Ye, Ruilong Li, Justin Kerr, Matias Turkulainen, Brent Yi, Zhuoyang Pan, Otto Seiskari, Jianbo Ye, Jeffrey Hu, Matthew Tancik, et al. gsplat: An open-source library for gaussian splatting. *JMLR*, 26(34):1–17, 2025. 2