Supplementary GeoGaussian: Geometry-aware Gaussian Splatting for Scene Rendering

Yanyan Li[1,2], Chenyu Lyu[3], Yan Di[2], Guangyao Zhai[2], Gim Hee Lee*[1], and Federico Tombari*[2,4]

[1] National University of Singapore, Singapore
[2] Technical University of Munich, Germany
[3] Tianjin University, China
[4] Google, Zurich, Switzerland

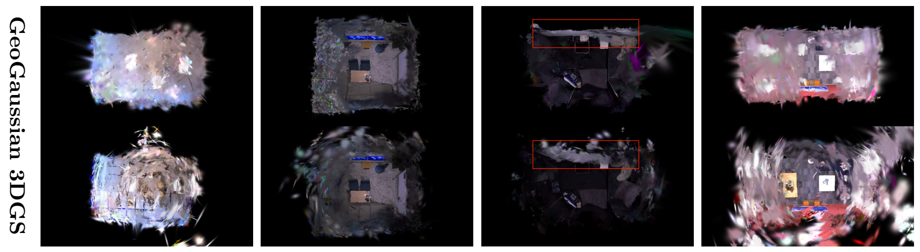
https://yanyan-li.github.io/project/gs/geogaussian

1 Novel View Rendering

In the evaluation section, we present a comprehensive set of experiments to evaluate the performance of our method. Here, we provide additional quantitative and qualitative results to further demonstrate the effectiveness of our approach.

1.1 Replica Sequences

In Table 2 (see Section 4.1), we compare novel view rendering performance using the R1, R2, OFF3, and OFF4 sequences. Additionally, we conduct experiments on four additional sequences (R0, OFF0, OFF1, and OFF2) in this section. Following the same setup as described in the section of *Gaussian Splatting in NVS* (see Section 4.4), we evaluate the rendering performance of 3DGS, LightGS, and GeoGaussian using three metrics, as shown in Table 5.



(a) R0 (100%) **(b)** OFF0(100%) **(c)** OFF1(100%) **(d)** OFF2(100%)

Fig. 6: Gaussian models generated by GeoGaussian (ours) and 3DGS on the Replica sequences.

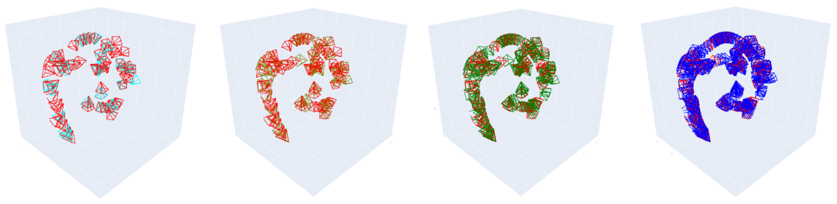
* Equal senior author

methods		3DGS				LightGS				GeoGaussian			
data		10%	16.6%	50%	100%	10%	16.6%	50%	100%	10%	16.6%	50%	100%
R0	PSNR↑	27.24	31.14	34.54	35.10	27.23	31.21	34.68	35.28	28.84	31.77	34.66	35.23
	SSIM↑	0.875	0.928	0.949	0.952	0.875	0.929	0.950	0.953	0.899	0.931	0.949	0.952
	LPIPS↓	0.070	0.040	0.031	0.032	0.070	0.040	0.031	0.031	0.057	0.036	0.028	0.029
OFF0	PSNR↑	31.61	35.99	41.70	42.67	31.63	36.00	41.82	43.08	32.42	36.34	42.11	42.74
	SSIM↑	0.927	0.959	0.979	0.981	0.927	0.959	0.979	0.981	0.924	0.960	0.980	0.981
	LPIPS↓	0.062	0.034	0.018	0.017	0.062	0.034	0.018	0.016	0.053	0.030	0.016	0.017
OFF1	PSNR↑	34.81	38.39	42.27	42.88	34.83	38.43	42.43	43.01	34.86	38.35	41.89	42.17
	SSIM↑	0.943	0.956	0.972	0.974	0.943	0.956	0.972	0.974	0.939	0.952	0.968	0.970
	LPIPS↓	0.060	0.046	0.036	0.035	0.060	0.045	0.036	0.035	0.061	0.045	0.039	0.039
OFF2	PSNR↑	28.27	32.91	36.72	37.34	28.25	32.85	36.95	37.49	29.52	33.97	36.90	37.32
	SSIM↑	0.911	0.948	0.964	0.966	0.911	0.948	0.965	0.967	0.925	0.955	0.968	0.970
	LPIPS↓	0.074	0.043	0.033	0.033	0.073	0.045	0.032	0.033	0.054	0.032	0.028	0.029
Avg.	PSNR↑	30.48	34.61	38.81	39.50	30.49	34.87	38.97	39.72	31.41	35.11	38.89	39.37
	SSIM↑	0.914	0.948	0.966	0.968	0.914	0.948	0.967	0.969	0.922	0.950	0.966	0.968
	LPIPS↓	0.067	0.041	0.030	0.029	0.066	0.041	0.029	0.029	0.056	0.036	0.028	0.029

Table 5: Comparison of rendering on the Replica dataset.

The 3D Gaussian models obtained by GeoGaussian and 3DGS are represented in Figure 6. Benefiting from the proposed densification method and geometric constraints, our method, GeoGaussian, preserves the reasonable geometry of environments. For example, the walls in Figure 6b and 6c are very thin, but the geometry is very noisy in the corresponding models of 3DGS.

Viewpoints in training and evaluation. In Table 2, we also evaluate the relationship between rendering performance and the sparsity of training views. Therefore, we visualize the position and orientation of viewpoints used in training and evaluation in Figure 7.

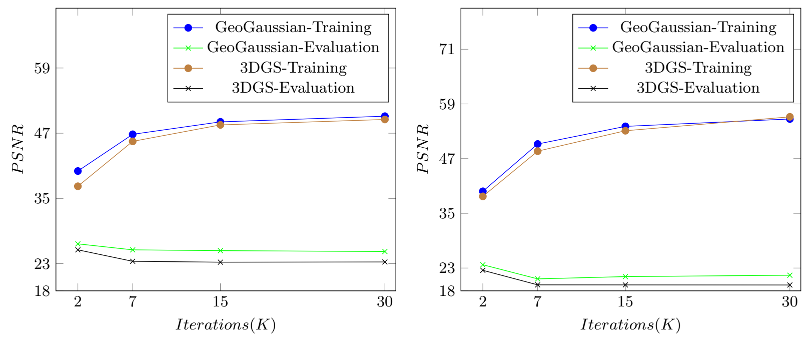


(a) OFF3 (10%) **(b)** OFF3 (16.6%) **(c)** OFF3 (50%) **(d)** OFF3 (100%)

Fig. 7: Viewpoints are used during training and evaluation, where the red camera is used for evaluation.

As shown in Figure 7, when the views used in training become sparse from OFF3(100%) to OFF3(10%), the scenarios seen from the training frames are difficult to cover from evaluation viewpoints. Therefore, the structure of the models plays an important role in maintaining the rendering performance as illustrated in Table 2 (see Section 4.4).

1.2 ICL-NUIM Sequences



(a) ICL-NUIM Room 2(10%) **(b)** ICL-NUIM Office 3(10%)

Fig. 8: Comparison of Rendering performance in training and evaluation datasets.

methods		3DGS				LightGS				GeoGaussian			
data		10%	16.6%	50%	100%	10%	16.6%	50%	100%	10%	16.6%	50%	100%
Room 1	PSNR↑	40.09	40.39	40.59	40.79	40.25	40.53	41.03	41.26	40.58	41.21	41.51	41.43
	SSIM↑	0.971	0.973	0.973	0.973	0.972	0.974	0.974	0.974	0.974	0.976	0.977	0.976
	LPIPS↓	0.022	0.023	0.024	0.025	0.021	0.022	0.023	0.023	0.019	0.019	0.019	0.019
Room 2	PSNR↑	23.31	30.15	37.33	39.10	23.28	30.18	37.51	39.23	25.23	31.26	37.70	39.46
	SSIM↑	0.777	0.906	0.965	0.974	0.775	0.906	0.965	0.974	0.841	0.925	0.968	0.975
	LPIPS↓	0.251	0.080	0.023	0.017	0.252	0.080	0.022	0.017	0.142	0.059	0.023	0.018
Office 2	PSNR↑	26.24	29.82	35.54	37.88	26.22	29.77	35.58	37.99	28.35	32.34	36.76	38.54
	SSIM↑	0.844	0.896	0.943	0.962	0.843	0.896	0.943	0.962	0.874	0.917	0.952	0.967
	LPIPS↓	0.145	0.078	0.036	0.024	0.146	0.078	0.035	0.023	0.100	0.055	0.026	0.017
Office 3	PSNR↑	19.28	22.86	32.20	36.04	19.19	22.82	32.21	36.06	21.42	27.06	33.52	36.19
	SSIM↑	0.718	0.848	0.950	0.975	0.716	0.847	0.949	0.975	0.769	0.907	0.966	0.977
	LPIPS↓	0.219	0.143	0.044	0.017	0.219	0.143	0.044	0.016	0.145	0.082	0.024	0.015
Avg.	PSNR↑	27.23	30.80	36.41	38.45	27.23	30.82	36.58	38.63	28.89	32.96	37.37	38.90
	SSIM↑	0.827	0.905	0.967	0.971	0.826	0.905	0.957	0.971	0.864	0.931	0.965	0.973
	LPIPS↓	0.159	0.081	0.031	0.020	0.159	0.087	0.031	0.019	0.101	0.053	0.024	0.017

Table 6: Comparison of rendering on the ICL-NUIM dataset.

As we mentioned in the Experiment section (see Section 4.2), the four sequences are used to evaluate these Gaussian Splatting approaches. As listed in Table 6, the proposed method shows robust rendering performance in different training settings compared with state-of-the-art Gaussian Splatting methods.

When the number of training frames is reduced, our method shows more robust performance since our models have better geometry to alleviate overfitting problems in rendering tasks. For example, our model trained on ICL-O3 (10%) achieves a PSNR of 21.42, while models of 3DGS and LightGS obtain 19.28 and 19.19, respectively.

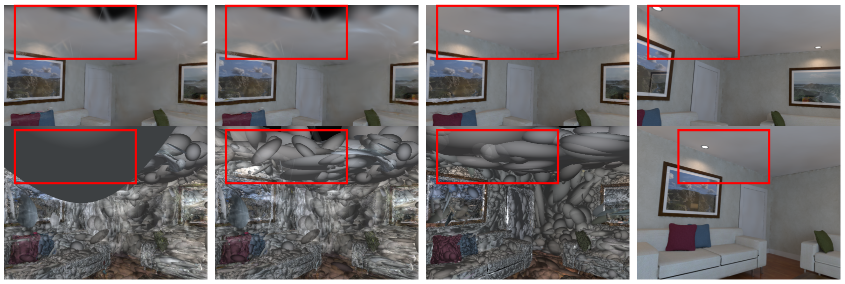


Fig. 9: Comparisons of novel view rendering on the ICL-NUIM datasets. In these scenarios, 3DGS and LightGS struggle with photorealistic rendering. As depicted in (d), the two training views surrounding the novel view.

As we mentioned in the Experiment section (see Section 4.2), the four sequences are used to evaluate these Gaussian Splatting approaches. As listed in Table 6, the proposed method shows robust rendering performance in different training settings compared with state-of-the-art Gaussian Splatting methods. When the number of training frames is reduced, our method shows more robust performance since our models have better geometry to alleviate overfitting problems in rendering tasks. For example, our model trained on ICL-O3 (10%) achieves a PSNR of 21.42, while models of 3DGS and LightGS obtain 19.28 and 19.19, respectively.

In an extreme case, when we observe the training process of these methods on ICL-O3 (10%), as shown in Figure 8, it is evident that Gaussian Splatting methods tend to overfit the training views due to the limited number of training views. However, our method maintains a certain level of generalization ability based on geometry constraints. For example, as the model improves its rendering results on the training dataset, the performance on the test dataset does not decrease significantly, as shown in Figure 8a.

1.3 TUM RGB-D Sequences

Structured environments from the TUM RGB-D dataset are utilized to evaluate rendering performance in Section 4.4. For a comprehensive evaluation of our method in general scenarios, we test it on the f3/lag-cabinet and f3/long-office sequences, which are non-structured Scenes, as shown in Table 7.

methods		3DGS				LightGS				GeoGaussian			
data		10%	16.6%	50%	100%	10%	16.6%	50%	100%	10%	16.6%	50%	100%
f3/lag-cabinet	PSNR↑	20.05	23.27	24.63	24.87	20.09	23.32	24.74	24.89	20.85	22.22	24.15	24.79
	SSIM↑	0.772	0.831	0.864	0.866	0.773	0.832	0.865	0.867	0.804	0.833	0.863	0.868
	LPIPS↓	0.226	0.143	0.123	0.127	0.225	0.142	0.123	0.126	0.178	0.158	0.130	0.125
f3/long-office	PSNR↑	16.67	18.88	23.54	24.33	16.69	18.93	23.66	24.43	18.89	20.55	24.56	25.12
	SSIM↑	0.598	0.686	0.812	0.832	0.600	0.689	0.813	0.835	0.643	0.736	0.833	0.843
	LPIPS↓	0.336	0.246	0.148	0.139	0.335	0.245	0.146	0.138	0.279	0.196	0.128	0.125
Avg.	PSNR↑	18.31	21.07	24.08	24.60	18.39	21.12	24.20	24.66	19.87	21.38	24.35	24.95
	SSIM↑	0.685	0.758	0.838	0.849	0.686	0.760	0.839	0.851	0.723	0.748	0.848	0.850
	LPIPS↓	0.281	0.194	0.135	0.133	0.280	0.193	0.134	0.132	0.228	0.177	0.129	0.125

Table 7: Comparison of rendering on the non-structured sequences on the TUM RGB-D dataset.

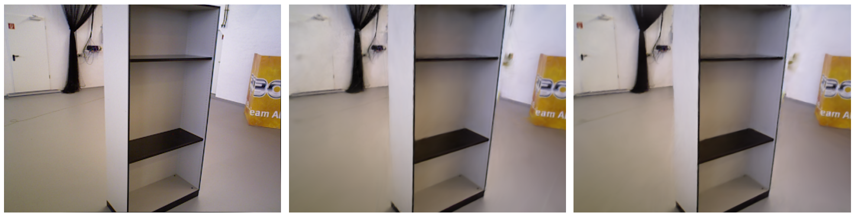

(a) Real scene **(b)** 3DGS **(c)** GeoGaussian(ours)

Fig. 10: Scene of the lag-cabinet sequence and render quality.

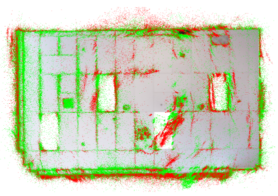
As mentioned in Section 4.4, the proposed method is not as effective as 3DGS and LightGS in the lag-cabinet sequence. This is because the far walls in this sequence cannot be accurately captured based on point clouds, making it challenging to obtain good normal vectors from the point clouds.

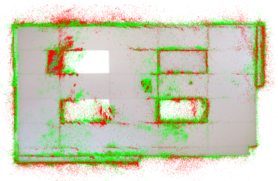
2 Model Reconstruction

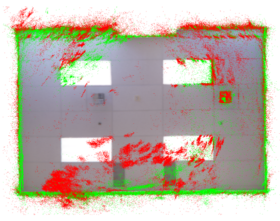
In this section, we present more quantitative results in the comparison of 3D Gaussian models. Based on the ground truth mesh models provided by the Replica dataset, we align these mesh models with point clouds from Gaussian models, where we randomly sample three points in each Gaussian ellipsoid.

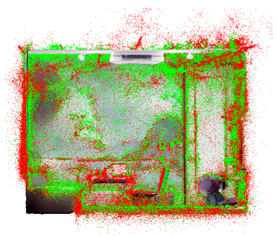
methods		R0	R1	R2	OFF0	OFF1	OFF2	OFF3	OFF4	Avg.
3DGS	mean	0.026	0.025	0.042	0.017	0.019	0.039	0.032	0.032	0.029
	std	0.066	0.081	0.146	0.050	0.055	0.201	0.066	0.112	0.097
GeoGaussian (ours)	mean	0.018	0.014	0.015	0.020	0.029	0.013	0.018	0.014	0.018
	std	0.032	0.016	0.028	0.042	0.067	0.024	0.020	0.023	0.031

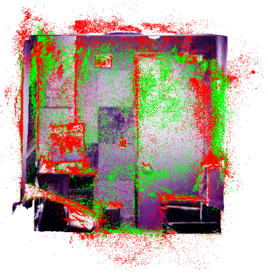
Table 8: Comparison of reconstruction performance on the Replica dataset.

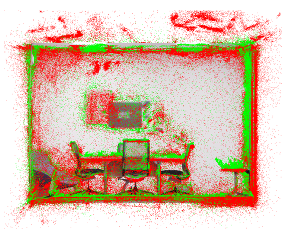

(a) R0

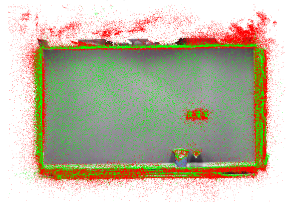

(b) R1

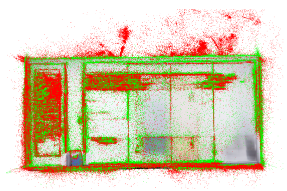

(c) R2


(d) OFF0


(e) OFF1


(f) OFF2


(g) OFF3


(h) OFF4

Ground truth mesh
Red points from 3DGS
Green points from GeoGaussian

(i) Description

Fig. 11: Reconstruction error visualization.

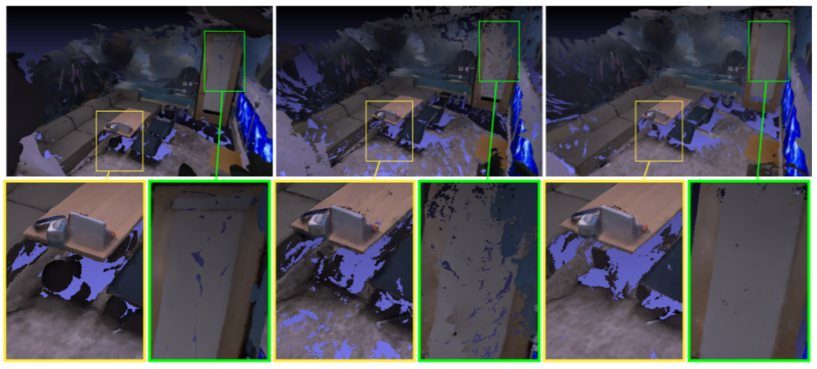


Fig. 12: Comparison in dense mesh modelling of Replica OFF0.

As illustrated in Table 8, the proposed method achieves better performance in mean and standard errors compared to 3DGS. Specifically, in the R2 sequence, the standard error of our method is 0.028, while the corresponding value for 3DGS is 0.146, which is 5 times worse than ours. Additionally, in OFF2, the mean error of 3DGS is 3 times greater than ours.

As shown in Figure 11, the point clouds from 3D Gaussians are aligned with mesh models, where the green and red points represent GeoGaussian and 3DGS, respectively. It is evident that points from our method align well with the model surface, indicating that our Gaussians are distributed according to the structure and texture of the training views. In contrast, the points from 3DGS are distributed more uniformly around the training perspective. This indicates that 3DGS is more inclined to perform densification along the angle of the training line of sight, while our method is more focused on training a highly versatile model rather than overfitting to the training perspectives.

In Figure 12, the mesh reconstruction results of 3DGS and our method are obtained by feeding Gaussian points into the Poisson reconstruction and texturing algorithms [3]. In low-textured regions, such as door and floor areas, the proposed method demonstrates robust and accurate performance in reconstructing dense models.

3 Performance in Wild Scenes and Limitations

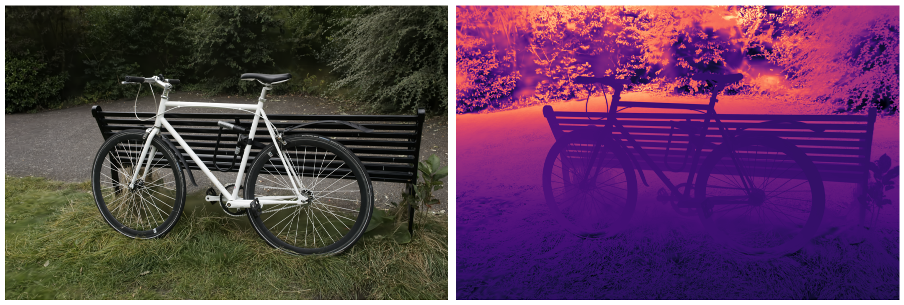


Fig. 13: Our depth render result in Mip-NeRF360 Bicycle.

To evaluate the performance of the proposed approach in wild scenes, additional popular datasets [1, 2] are used to test GeoGaussian and 3DGS in this section. As shown in Figure 13 and 14, we compare our method with 3DGS in wild scenarios, demonstrating that GeoGaussian is more robust and converges faster since our method requires only local smooth regions instead of large plane features. In 3DGS, 3D Gaussians are optimized via photometric residuals, resulting in floaters in texture-less regions. As shown in Figure 14, while the rendering result of 3DGS is acceptable, the 3D geometry is clearly not well reconstructed.

This phenomenon has been witnessed in other indoor sequences (see Figure 6 and 11).

To solve the problem, our system introduces geometric constraints in the initialization, densification, and optimization modules. This design enhances the quality of scene geometry, leading to robust rendering performance (Figure 3). However, there are limitations to this approach, summarized as follows:

- The geometric constraints degenerate if few smooth regions are detected;
- Incorrect geometric residuals can occur when fake smooth regions are initialized as thin Gaussians, although the effect of outliers is controllable.

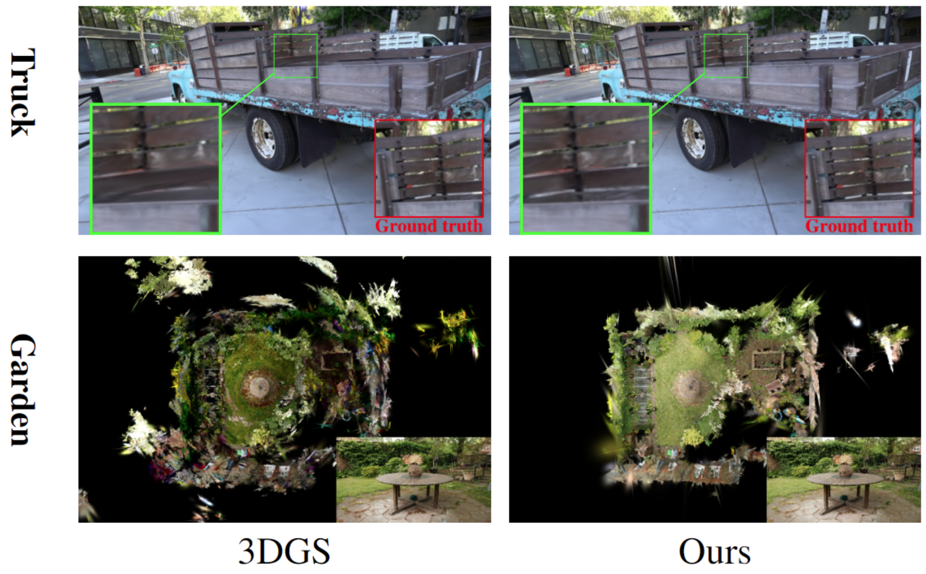
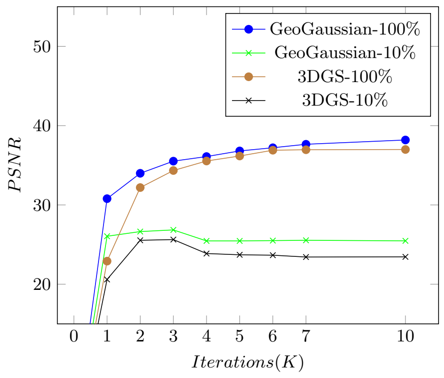
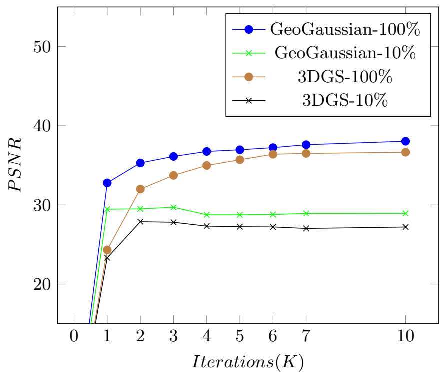


Fig. 14: 2D and 3D results in Truck [2] and Garden scenes [1].

4 Initialization and Densification



(a) ICL-NUIM Room 2



(b) ICL NUIM Office 2

Fig. 15: Rendering performance in the first 10,000 iterations.

In Figure 15, the rendering performance of 3DGS and GeoGaussian approaches in the first 10,000 iterations is presented. Our method stops increasing Gaussians after 10,000 iterations, while 3DGS continues to increase Gaussians until 15,000 iterations. Therefore, the densification progress mainly occurs in the first 10,000 iterations, where Gaussians are initialized and dramatically densified under the supervision of photometric and geometric constraints.

As shown in Figure 15, the PSNR performance increases rapidly in the first 10,000 iterations for both 3DGS and GeoGaussian methods. But, our method is easy to train, especially in sparse view rendering tasks like ICL-NUIM Room 2(10%) and ICL-NUIM Office 2(10%).

References

1. Barron, J.T., Mildenhall, B., Verbin, D., Srinivasan, P.P., Hedman, P.: Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5470–5479 (2022)
2. Knapitsch, A., Park, J., Zhou, Q.Y., Koltun, V.: Tanks and temples: Benchmarking large-scale scene reconstruction. ACM Transactions on Graphics (ToG) **36**(4), 1–13 (2017)
3. Ye, C., Nie, Y., Chang, J., Chen, Y., Zhi, Y., Han, X.: Gaustudio: A modular framework for 3d gaussian splatting and beyond. arXiv preprint arXiv:2403.19632 (2024)