Improving Densification in 3D Gaussian Splatting for High-Fidelity Rendering

Supplementary Material

A. Quantitative Comparison per Scene

Scene—Methods	3DGS	AbsGS	PixelGS	MiniGS	TamGS	MCMC	SteepGS	PercepGS	Ours
bicycle	0.76561	0.78243	0.77857	0.79877	0.77766	0.79896	0.73151	0.79283	0.80369
flowers	0.60522	0.62010	0.63628	0.64182	0.61695	0.64659	0.54756	0.63873	0.63902
garden	0.86670	0.86803	0.87039	0.87853	0.87378	0.87676	0.85584	0.87014	0.88139
stump	0.77057	0.78059	0.78454	0.80582	0.77425	0.81188	0.73186	0.79787	0.80930
treehill	0.63369	0.62100	0.63534	0.64207	0.64464	0.65881	0.61224	0.64299	0.65686
bonsai	0.94217	0.94456	0.94657	0.94805	0.94532	0.94784	0.93827	0.94786	0.94911
counter	0.90860	0.91116	0.91465	0.91044	0.91327	0.91709	0.90180	0.91520	0.91909
kitchen	0.92771	0.93075	0.93117	0.93344	0.93180	0.93339	0.92294	0.93130	0.93617
room	0.91915	0.92565	0.92242	0.92808	0.92460	0.92954	0.91558	0.92826	0.93194
playroom	0.90743	0.90871	0.90571	0.90753	0.90612	0.91606	0.90630	0.90715	0.91587
drjohnson	0.90096	0.90194	0.88791	0.90505	0.90801	0.90809	0.90279	0.90454	0.91055
train	0.81321	0.82877	0.82760	0.82125	0.82614	0.83955	0.80120	0.82613	0.84640
truck	0.88184	0.88549	0.88693	0.88939	0.89293	0.89926	0.87559	0.88798	0.89837

Table 1. The SSIM scores for all works in each scene.

Scene—Methods	3DGS	AbsGS	PixelGS	MiniGS	TamGS	MCMC	SteepGS	PercepGS	Ours
bicycle	25.213	25.372	25.279	25.581	25.504	25.681	24.798	25.527	25.866
flowers	21.539	21.368	21.580	21.526	21.871	22.010	20.713	21.494	21.768
garden	27.361	27.408	27.493	27.693	27.898	27.811	27.125	27.628	28.098
stump	26.539	26.726	26.843	27.140	26.632	27.384	25.832	27.030	27.211
treehill	22.495	22.094	22.296	22.234	23.024	22.944	22.204	22.408	22.839
bonsai	32.242	32.145	32.547	32.163	32.889	32.646	31.806	32.607	32.966
counter	29.016	29.096	29.181	28.560	29.486	29.348	28.743	29.288	29.798
kitchen	31.474	31.852	31.752	31.704	32.131	32.040	30.919	31.909	32.554
room	31.446	31.634	31.588	31.528	32.199	32.188	31.330	32.040	32.652
playroom	30.019	30.051	29.876	30.447	30.186	30.449	30.099	30.200	30.627
drjohnson	29.119	28.933	28.084	29.404	29.670	29.112	29.385	29.551	29.763
train	21.958	22.126	22.145	21.320	22.780	22.463	21.678	22.180	22.641
truck	25.414	25.535	25.543	25.409	26.062	26.338	25.169	25.572	26.542

Table 2. The PSNR scores for all works in each scene.

Scene—Methods	3DGS	AbsGS	PixelGS	MiniGS	TamGS	MCMC	SteepGS	PercepGS	Ours
bicycle	0.20921	0.18275	0.17956	0.15755	0.19210	0.16815	0.26239	0.17383	0.16510
flowers	0.33536	0.28512	0.26170	0.25465	0.33052	0.28063	0.39758	0.26712	0.28529
garden	0.10651	0.10663	0.09873	0.09011	0.09846	0.09575	0.12678	0.10273	0.09222
stump	0.21659	0.20442	0.18780	0.16817	0.20476	0.17079	0.27371	0.18501	0.17718
treehill	0.32461	0.29297	0.27505	0.26107	0.31206	0.27054	0.37381	0.28248	0.28717
bonsai	0.20380	0.19269	0.19116	0.17372	0.19937	0.18995	0.21073	0.18098	0.18503
counter	0.19967	0.19324	0.18275	0.17333	0.19408	0.18346	0.21453	0.17634	0.17865
kitchen	0.12570	0.12130	0.11883	0.11396	0.12117	0.12040	0.13437	0.11704	0.11389
room	0.21834	0.20404	0.20970	0.18750	0.20872	0.19769	0.22849	0.19437	0.19277
playroom	0.24304	0.24314	0.24037	0.20347	0.23718	0.23554	0.25294	0.23122	0.22560
drjohnson	0.24406	0.24378	0.25469	0.21834	0.23513	0.23779	0.24825	0.23090	0.22588
train	0.20741	0.18948	0.17783	0.17908	0.20080	0.18502	0.22766	0.18306	0.16947
truck	0.14644	0.13846	0.12024	0.10031	0.12577	0.11215	0.15890	0.11761	0.12091

Table 3. The LPIPS scores for all works in each scene.

B. Implementation Details

We built our code upon the TamingGS repository. In addition to the proposed method, we made the following modifications:

- Regarding hyperparameters, we adjusted position lr init to 0.00004 and position lr final to 0.000002. The gradient threshold used in EAS is set to 0.0003 and the opacity reset threshold is set 0.05.
- We removed the pruning of large Gaussians, as our method can correctly identify these Gaussians and optimize their covered regions through splitting.

C. More Qualitative Comparisons

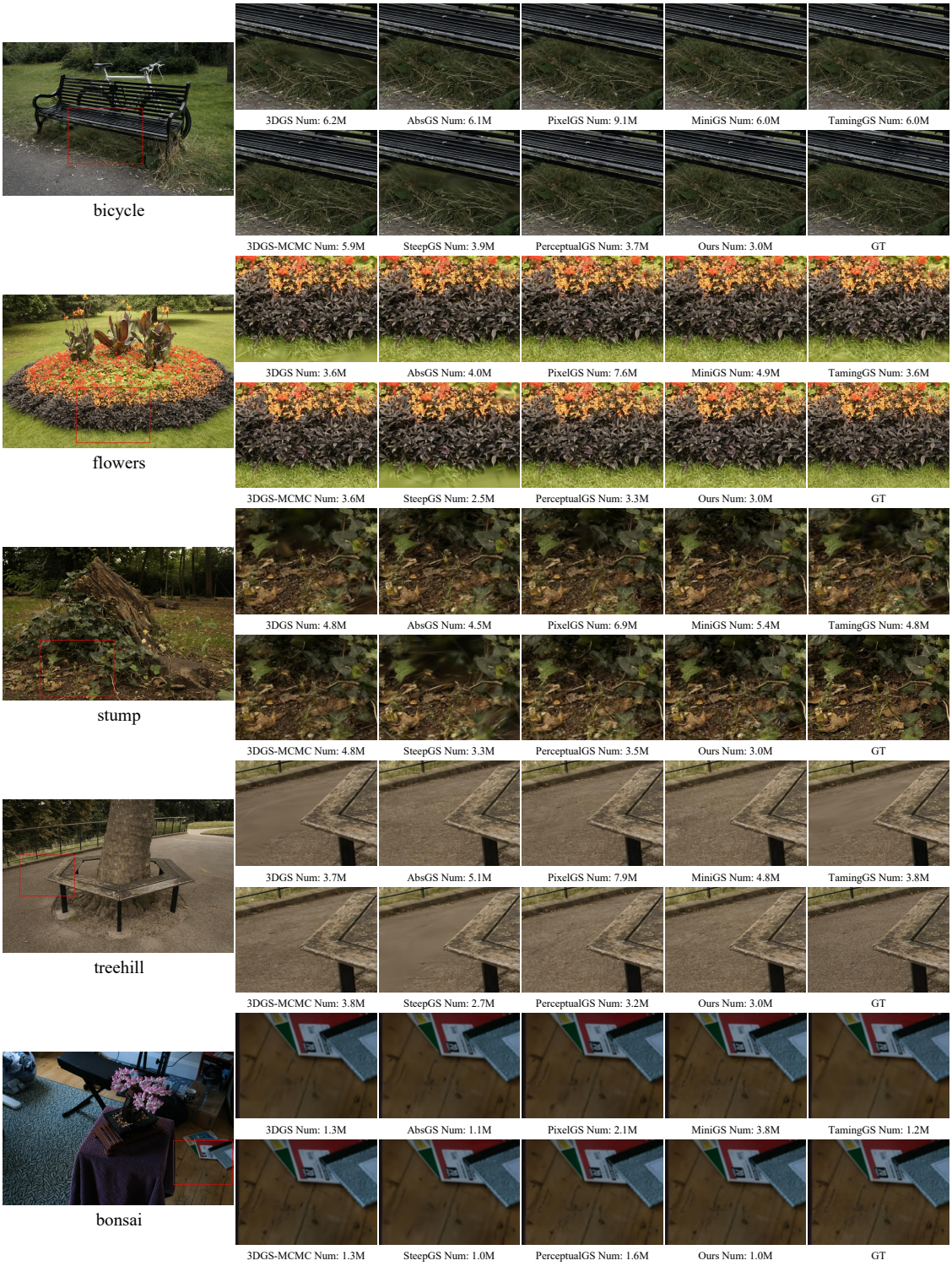

Figure 1. Qualitative comparison results among scenes bicycle, flowers, stump, treehill, bonsai.

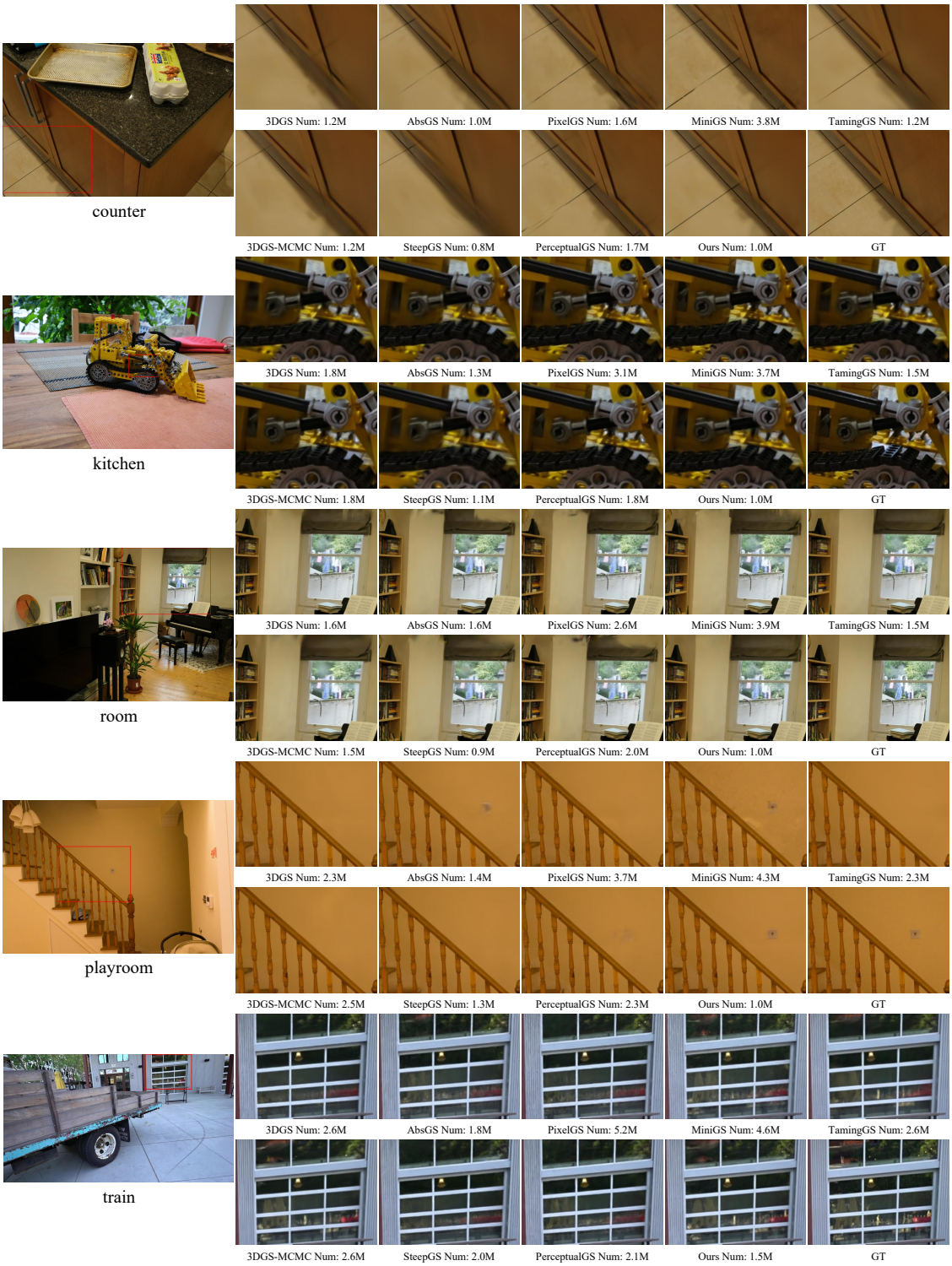

Figure 2. Qualitative comparison results among scenes counter, kitchen, room, playroom, truck.

D. Proof Regarding the Optimal Rs

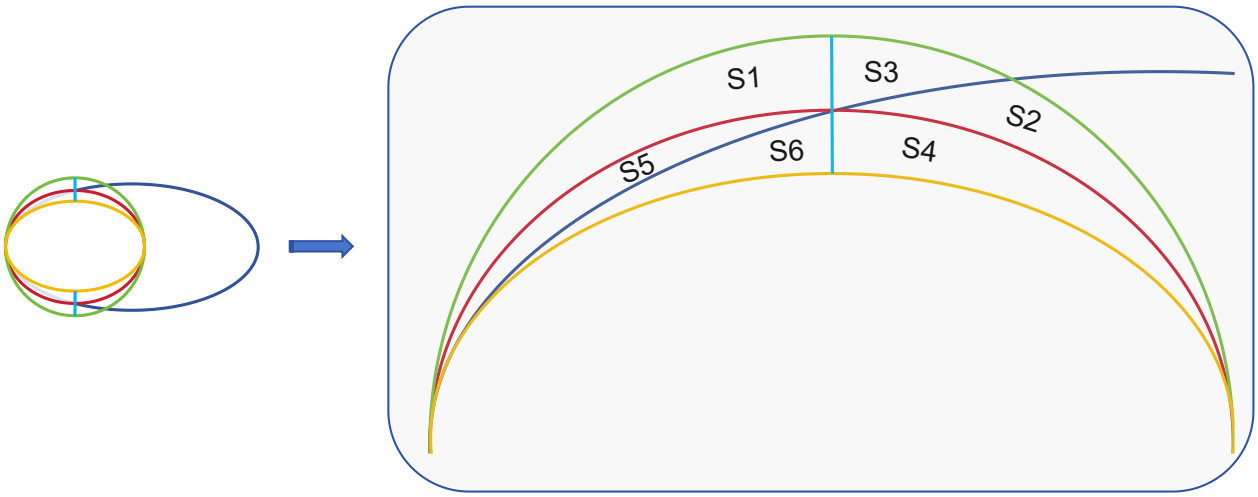

Figure 3. Schematic illustration of the split operation when simplified to a 2D ellipse.

To analyze the geometric disturbance introduced by the split operation, the shape change of a 3D Gaussian is projected onto the 2D cross-section orthogonal to its longest axis. Since the two minor axes share the same scaling ratio during initialization, the cross-section can be approximated as a circle of radius R_0, and the child Gaussian becomes a circle of radius R_s whose center is displaced by a distance d along the longest axis (see Figure 3). The goal is to choose R_s such that the geometric discrepancy between child and parent Gaussians is minimized. This discrepancy is measured as the difference

$$S_{\text{overlap}} - S_{\text{non-overlap}},$$

where S_{overlap} is the overlap area and $S_{\text{non-overlap}}$ is the sum of regions that do not overlap.

Tangency as the optimality condition. The geometric discrepancy is minimized when the two circles are locally tangent to each other at the intersection chord. A necessary and sufficient condition for tangency between a parent circle of radius R_0 and a child circle of radius R_s, whose center is displaced by d, is

$$R_s^2 = R_0^2 - d^2.$$

Rewriting this using the normalization along the parent's longest axis L_0 yields the desired expression:

$$R_s = R_0\sqrt{1 - \frac{d^2}{L_0^2}}. \tag{1}$$

At this configuration, the endpoints of the child Gaussian's minor axes lie exactly on the parent's surface, producing the maximum-length intersection chord and therefore the smallest local geometric deviation.

Area variation under radius perturbation. To verify that (1) indeed gives a maximum, consider a small perturbation $R_s \to R_s + \delta R$.

When $\delta R > 0$, the child Gaussian expands. Let S_2 denote the additional overlapping region, and S_1 and S_3 denote the two newly emerged non-overlapping regions. A geometric identity at the intersection chord ensures

$$S_1 = S_2 + S_3.$$

Thus the change in the discrepancy measure is

$$\Delta\left(S_{\text{overlap}} - S_{\text{non-overlap}}\right) = S_2 - (S_1 + S_3) = -2S_2 < 0.$$

Therefore the discrepancy strictly decreases for any increase in R_s.

When $\delta R < 0$, the child Gaussian shrinks. Let S_5 denote the reduced non-overlapping region and $S_4 + S_6$ denote the reduced overlapping region. The symmetric geometric identity gives

$$S_5 = S_4 + S_6.$$

Thus,

$$\Delta\left(S_{\text{overlap}} - S_{\text{non-overlap}}\right) = -(S_4 + S_6) - (-S_5) = -2S_6 < 0.$$

Again, the discrepancy strictly decreases for any deviation in R_s.

Conclusion. Both enlarging and shrinking R_s lead to a strictly smaller value of $S_{\text{overlap}} - S_{\text{non-overlap}}$, therefore the expression in (1) is a maximizer of this measure and yields the minimal geometric disturbance between the parent and child Gaussians.

Note that this analysis focuses on geometric consistency. In practice, the rendering contribution is also affected by opacity, color, and interactions with nearby Gaussians. However, because contributions near Gaussian boundaries are inherently weak, this geometric solution remains sufficiently close to optimal for densification in 3D Gaussian Splatting. Figure 4 provides a comparison of results on the "stump" scene using different values of R_s.

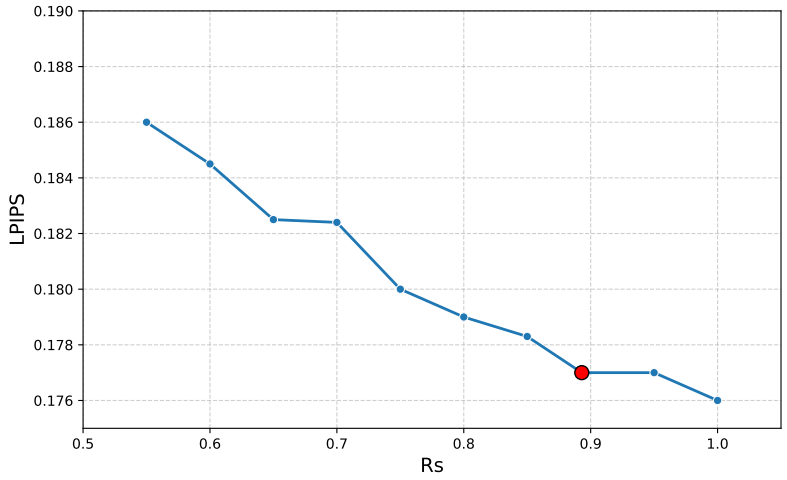

Figure 4. The trend of LPIPS with respect to Rs in the stump scene. The red dot indicates the value we chose in practice.

E. Quantitative Comparison of Training View

Dataset	Mip-NeRF360				Deep Blending				Tanks&Temples			
Method—Metric	$SSIM^{\uparrow}$	$PSNR^{\uparrow}$	$LPIPS^{\downarrow}$	$Num^{\downarrow}$	$SSIM^{\uparrow}$	$PSNR^{\uparrow}$	$LPIPS^{\downarrow}$	$Num^{\downarrow}$	$SSIM^{\uparrow}$	$PSNR^{\uparrow}$	$LPIPS^{\downarrow}$	$Num^{\downarrow}$
3DGS	0.88763	29.478	0.17436	3337659	0.95566	36.910	0.19072	2832494	0.88884	26.349	0.15846	1847041
AbsGS (ACMMM24)	0.90764	29.990	0.15170	3194225	0.95449	36.645	0.19275	2054043	0.89903	26.572	0.14515	1442984
PixelGS (ECCV24)	0.91821	30.463	0.13830	5619828	0.95783	36.934	0.18320	4644260	0.91278	27.015	0.12096	4519926
MiniSplatting-D (ECCV24)	0.91928	30.271	0.12731	4685127	0.96100	37.080	0.14321	4627579	0.91234	26.708	0.10628	4260423
TamingGS (SIGGRAPHAsia24)	0.89414	30.071	0.16662	3182444	0.95643	37.123	0.19100	2799868	0.89699	27.362	0.14630	1849918
3DGS-MCMC (NeurIPS24)	0.90720	30.227	0.15125	3227778	0.94561	34.791	0.20561	2950000	0.90585	26.923	0.13067	1850000
SteepGS (CVPR25)	0.85837	28.454	0.21009	2193808	0.95122	36.394	0.20307	1605267	0.87584	25.815	0.17705	1310323
Perceptual-GS (ICML25)	0.90476	30.046	0.14792	2685908	0.96035	37.478	0.17241	2892183	0.90570	26.732	0.12616	1721090
Ours	0.91215	30.930	0.14560	1777778	0.96014	38.327	0.17935	1250000	0.91639	28.048	0.12316	1250000

Table 4. Quantitative results of trainging view.

Although our primary focus lies on the rendering quality of test viewpoints, the quality of training viewpoints is also critically important in practical applications. As shown in Table 4, which presents the training viewpoint scores of all compared methods, our approach still demonstrates a substantial advantage.

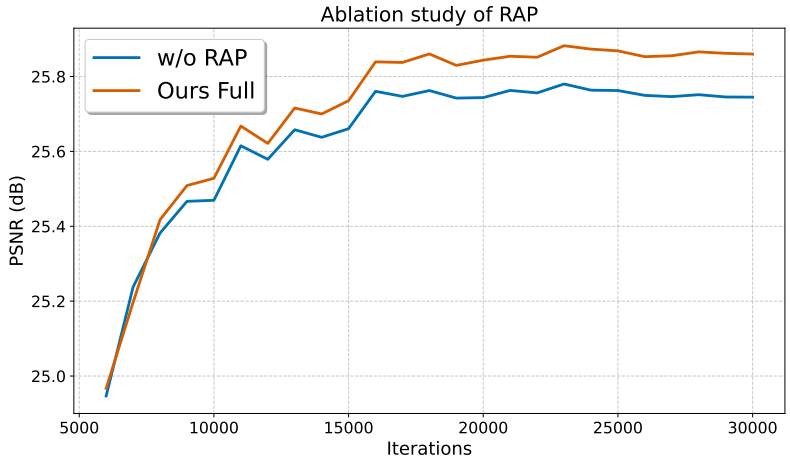

Figure 5. The impact of RAP on bicycle scene.

F. Detailed Analysis of RAP

The PSNR–iteration curves in Figure 5 compare optimization dynamics with and without RAP. With RAP enabled, the rendering quality consistently stays ahead and exhibits a stable improvement margin. This indicates that RAP effectively removes overfitted Gaussians whose overall contribution to rendering quality is negative.

Methods—Scene	bicycle	flowers	garden	stump	treehill	bonsai	counter	kitchen	room	average
wo RAP	25.715	21.669	28.011	27.051	22.768	32.360	29.809	32.588	32.655	28.069
Ours Full	25.866	21.768	28.098	27.211	22.839	32.966	29.802	32.564	32.652	28.196
Difference	0.151	0.100	0.087	0.160	0.071	0.606	-0.007	-0.025	-0.003	0.127

Table 5. The SSIM scores for all works in each scene.

Table 5 reports per-scene quantitative results. RAP does not introduce noticeable degradation in any scene and yields considerable improvements in many of them. It is worth noting that the *bonsai* scene has a high probability of producing floaters without RAP, but this issue does not occur deterministically. The values of bonsai in Table 5 reflect the fact that floaters happened to appear in all three runs without RAP, resulting in lower PSNR. The same observation applies to the *train* scene.

For the *bicycle* scene, we further analyzed the Gaussians pruned by the first RAP. Their average opacity ranking before and after RAP dropped from 14.5% to 10%, respectively. This suggests that a noticeable portion of the removed Gaussians had relatively slow opacity recovery, causing them to rank lower and thus be identified by RAP as redundant or floater-like components.

The choice of RAP hyperparameters is based on extensive empirical testing across all scenes. We apply RAP twice because two rounds consistently outperform a single round, while additional rounds bring no further improvement. We also execute RAP after the first two resets rather than after later ones, as earlier pruning leaves more total optimization time for newly added Gaussians and yields higher final quality.

The recovery interval after each reset must ensure both sufficient opacity recovery and adequate coverage across viewing directions. An interval of 300 iterations strikes a good balance between these requirements. The pruning ratio of 20% is also empirically determined; higher ratios increase floater suppression but simultaneously risk removing useful Gaussians. A 20% threshold provides a reliable balance between the two.

G. More Ablation Study

Figure 6 illustrates the optimization speed comparison between our method and 3DGS in the bicycle scene. It can be observed that our method not only significantly outperforms 3DGS in final rendering quality but also achieves faster overall convergence speed.

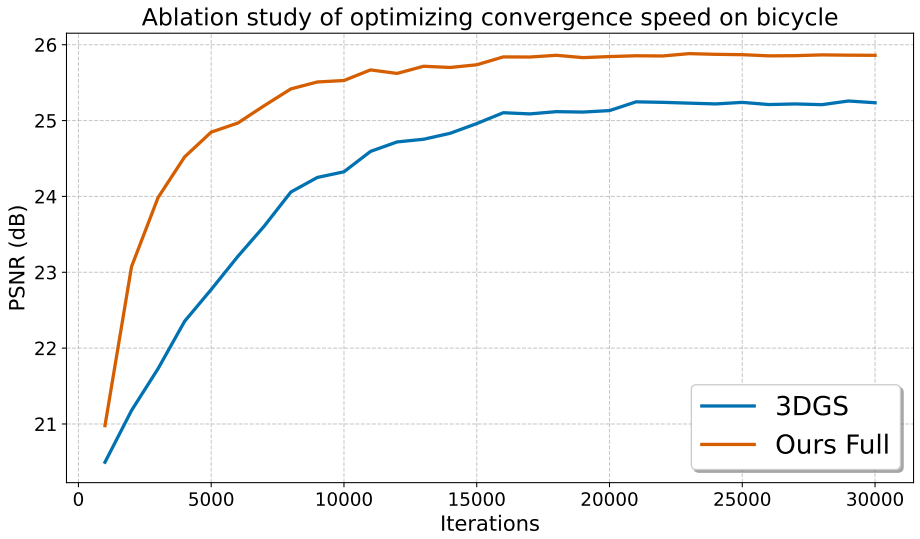

Figure 6. Ablation study of optimizing convergence speed on bicycle.

Figure 7 shows the impact of enabling MU at different times. According to the figure, enabling MU from the beginning significantly degrades rendering quality. The negative impact of MU on rendering quality gradually diminishes as the densification process progresses. Enabling MU at the end of the densification process can improve rendering quality. Our two-stage MU approach performs slightly better than the single-stage MU.

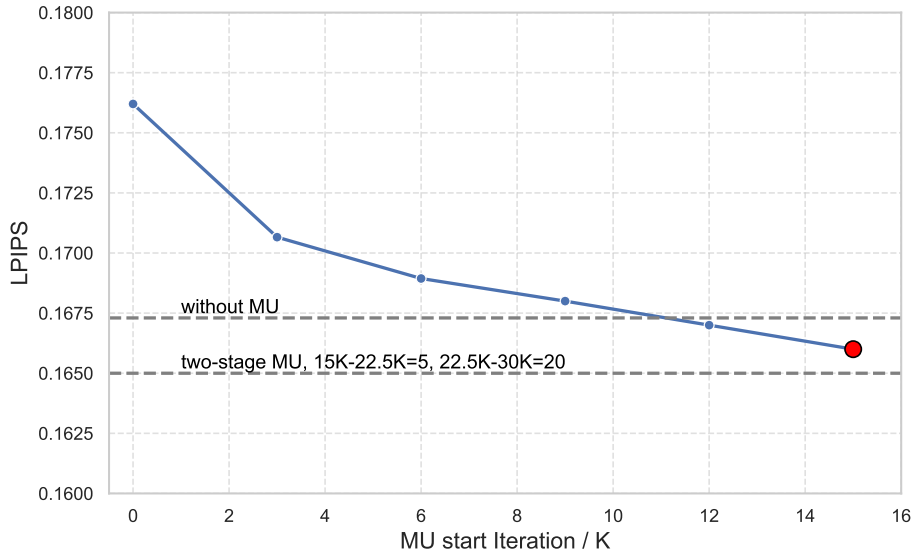

Figure 7. The impact of different MU starting iterations on LPIPS, test scene is bicycle. The update interval for parameters is uniformly set to 5. The red dot indicates the value we chose in practice.

Figure 8 shows the impact of different Opacity Reduction Rates on rendering quality. As can be seen from the figure, reducing opacity after splitting improves rendering quality, regardless of the reduction magnitude. When the Rate varies between 0.4 and 0.8, the change in rendering quality is relatively small. We choose 0.6 as an empirically determined optimal value.

Figure 9 shows the impact of the distance (d) between child Gaussians and the original Gaussian center after splitting, on both rendering quality and FPS. As d gradually decreases from 0.5, the geometric discrepancy caused by splitting slightly

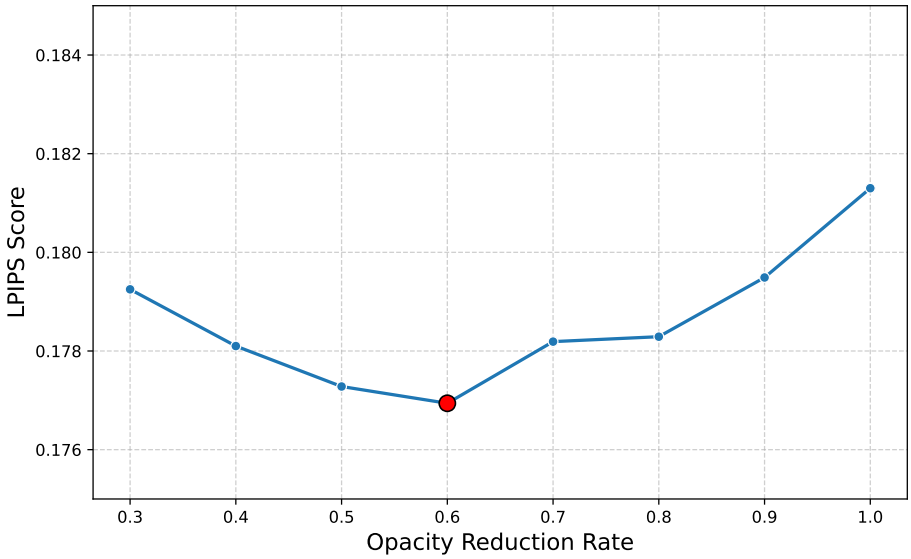

Figure 8. The trend of LPIPS with respect to opacity reduction rate in the stump scene. The red dot indicates the value we chose in practice.

reduces, but the overlapping area among Gaussians increases significantly. This overlap increases the pixel rendering queue length, leading to higher rendering costs. Moreover, we observe that as d decreases, the negative effect of Gaussian overlap on rendering quality gradually outweighs the positive effect from reduced geometric discrepancy. The value of 0.45 we choose is an empirically determined trade-off that balances FPS and rendering quality.

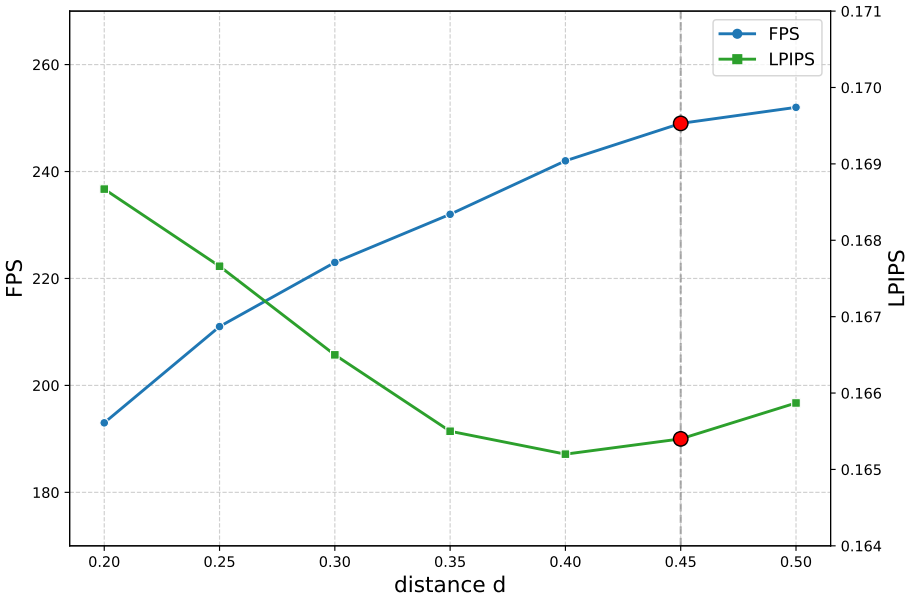

Figure 9. Testing the impact of the distance d between the sub-Gaussian and the original Gaussian centers on quality and rendering frame rate, test scene is bicycle.

Figure 10 shows the growth curves of Gaussians with and without GC. Without GC, the number of Gaussians reaches its peak early in the densification process, which clearly increases rendering overhead. In contrast, with GC applied, the growth of Gaussians follows a smoother trend, reaching its peak precisely at the end of densification.

Figure 11 illustrates the quality degradation caused by each splitting operation in LAS compared to split. A smaller

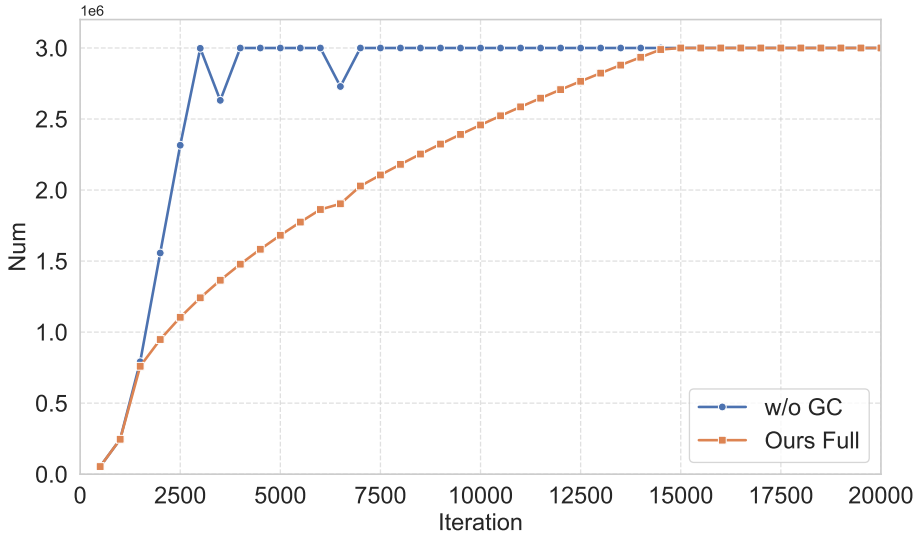

Figure 10. Growth curves of Gaussians with and without GC, test scene is bicycle.

quality loss indicates that the geometric error introduced by densification itself is smaller. The results shown in the figure demonstrate that LAS can significantly reduce the geometric error induced by splitting.

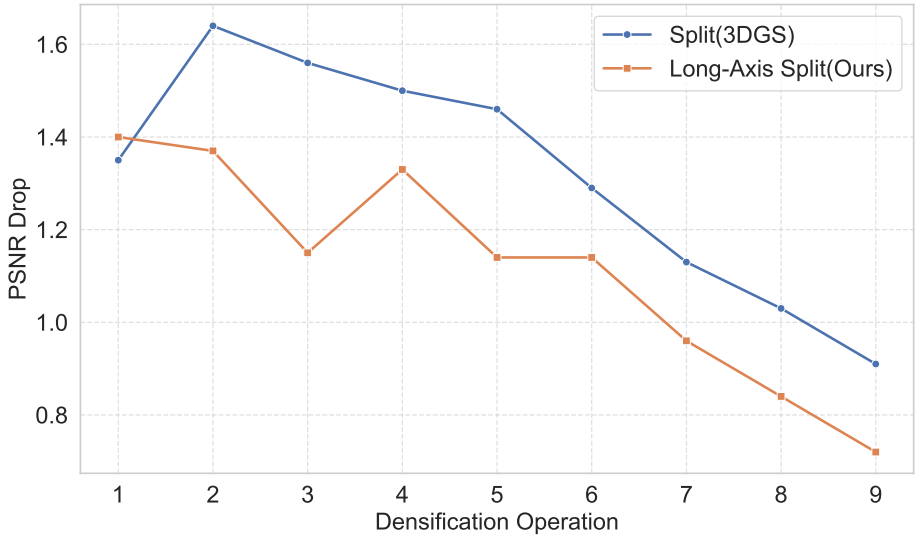

Figure 11. Evaluate the drop in PSNR after splitting using different splitting strategies, test scene is bicycle, a smaller drop indicates less geometric error introduced by the splitting.