Urban-GS: A Unified 3D Gaussian Splatting Framework for Compact and High-Fidelity Aerial-to-Street Reconstruction

Supplementary Material

1. More Implementation details

In this section, we provide more detailed implementation settings of our method. We follow Horizon-GS [1] to construct a LoD-structured representation from the SfM-derived point cloud of the scene. Specifically, to avoid generating an excessive number of anchors during initialization, we limit the maximum LoD level to 6. The initial scale of each anchor is set to the voxel size of its corresponding level. The loss function parameters are λ_{ssim}=0.2, λ_{vol}=0.01, λ_o=0.05 and λ_m=0.003. The λ_d is exponentially decayed from 1 to 0.01. Depth supervision is activated after 500 iterations. For the global training stage, mask supervision is activated after 3,000 iterations. Specifically, only anchors that are observed more than 10 times are allowed to participate in mask supervision, in order to prevent newly generated anchors from being prematurely removed. In the local refinement stage, mask supervision is applied from the beginning, with the mask score attribute inherited from the global training phase. Each anchor is sampled ten times, and anchors with all corresponding M_a values equal to zero are removed. This pruning procedure is executed at every densification step and subsequently every 2,000 iterations after densification.

2. Additional Experiments and Results

2.1. Additional Ablations Study and Analysis

Per-Scene Ablation Experiment Results. In the main text, we used the synthetic scenes CitySample and Elvenruin, along with the real-world Road scene, for ablation discussions. This section presents the detailed results for each scene. The detailed results are presented in Tab. 1, 2, 3 and 4.

Effect of γ_{scale}. In the Contribution-based Anchor Pruning strategy, we apply a weighting factor γ_{scale} to the per-view contribution w_i^v of each neural Gaussian ng_i. This design is motivated by our empirical observations. As shown in Fig. 1, the distribution of w_i concentrates around relatively low values, with the highest contributions typically appearing near 0.3. This behavior is expected: according to the rendering formulation of 3DGS [2], a Gaussian contributes more strongly to pixels near its projected center and much less to pixels near the boundary. As a result, averaging contributions across all pixels naturally leads to lower per-view contribution values. To ensure that these high-contribution neural Gaussians are properly preserved during pruning, we introduce the scaling factor γ_{scale}. Its effect is summarized in Tab. 5, where we observe that setting γ_{scale} to 2 or 3 provides the most balanced trade-off between retaining important anchors and removing low-contribution ones.

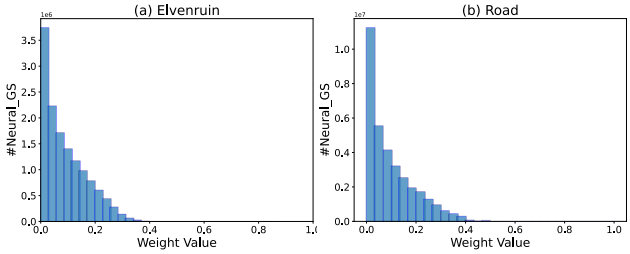

Figure 1. The distribution of w_i collected from scene *Elvenruin* (a) and scene *Road* (b).

Effect of λ_m. In this section, we analyze the impact of varying the mask-loss weight λ_m on the pruning results, as summarized in Tab. 6. As λ_m increases, a larger number of anchors are removed due to the stronger pruning constraint. As λ_m increases from 0.001 to 0.003, the quality of novel-view rendering remains largely stable, while the number of anchors is effectively reduced. However, when λ_m is further increased to 0.004, a noticeable degradation in reconstruction quality emerges. To achieve a balanced trade-off between reconstruction quality and modeling efficiency, we adopt $\lambda_m = 0.003$ as our final setting.

Comparison of Global-to-Local Optimization Iteration Settings. In this section, we compare two iteration configurations for the Global-to-Local Optimization stage. The first configuration applies local densification using the unstable-view groups for the initial 10k iterations, followed by another 10k iterations with uniform view sampling and no further densification. The second configuration performs densification using the unstable-view groups throughout the entire 20k iterations. As shown in Tab. 7, we empirically observe no significant performance gap between the two configurations. For fairness, we adopt the 10k + 10k setting in our main experiments, ensuring that the total number of iterations remains consistent with other baseline methods (i.e., 50k iterations in total).

Additional visualization results To better illustrate the effectiveness of our method, we provide additional visual results, including our reconstructed scenes and comparisons against Horizon-GS [1]. Please refer to *demo.mp4* for details.

Scene	Elvenruin				Road				Citysample			
Method|Metrics	**PSNR ↑**	**SSIM ↑**	**LPIPS ↓**	**Anchors ↓**	**PSNR ↑**	**SSIM ↑**	**LPIPS ↓**	**Anchors ↓**	**PSNR ↑**	**SSIM ↑**	**LPIPS ↓**	**Anchors ↓**
Baseline	27.93	0.880	0.155	1815k	21.34	0.647	0.404	3671k	26.33	0.864	0.213	2836k
+AJAD	28.36	0.894	0.122	4292k	21.57	0.686	0.316	9832k	27.05	0.880	0.188	15015k
+CAP	28.23	0.890	0.135	1901k	21.41	0.680	0.320	2742k	26.85	0.876	0.194	3714k
+GLO	28.78	0.899	0.128	1856k	21.72	0.691	0.312	2712k	27.66	0.886	0.185	3480k

Table 1. **Per-scene ablation on main model components.** "+" means add components on basis of all components in the above rows. "AJAD", "CAP", and "GLO" denote our proposed Aerial–Street Joint Adaptive Densification, Contribution-based Anchor Pruning, and Global-to-Local Optimization, respectively.

Scene	Elvenruin				Road				Citysample			
Method|Metrics	**PSNR ↑**	**SSIM ↑**	**LPIPS ↓**	**Anchors ↓**	**PSNR ↑**	**SSIM ↑**	**LPIPS ↓**	**Anchors ↓**	**PSNR ↑**	**SSIM ↑**	**LPIPS ↓**	**Anchors ↓**
Base [2]	27.93	0.880	0.155	1815k	21.34	0.647	0.404	3671k	26.33	0.864	0.213	2836k
w/ Hier-GS [3]	28.28	0.891	0.128	3964k	21.43	0.678	0.321	9317k	26.71	0.871	0.189	12610k
w/ Abs-GS [5]	27.97	0.881	0.153	1807k	21.28	0.672	0.323	4054k	26.51	0.865	0.211	2931k
w/ Ours	28.36	0.894	0.122	4292k	21.57	0.686	0.316	9832k	27.05	0.880	0.188	15015k

Table 2. **Per-scene comparison results of different densification strategies.** This experiment provides an extended analysis of the "+AJAD" results presented in Tab. 1. Here, "Base" refers to the original densification triggering strategy used in 3DGS [2].

Scene	Elvenruin				Road				Citysample			
Method|Metrics	**PSNR ↑**	**SSIM ↑**	**LPIPS ↓**	**Anchors ↓**	**PSNR ↑**	**SSIM ↑**	**LPIPS ↓**	**Anchors ↓**	**PSNR ↑**	**SSIM ↑**	**LPIPS ↓**	**Anchors ↓**
Base	28.36	0.894	0.122	4292k	21.57	0.686	0.316	9832k	27.05	0.880	0.188	15015k
Global($\lambda_m = 0.001$)	28.16	0.890	0.136	2178k	21.30	0.678	0.324	3035k	26.74	0.874	0.193	4464k
Ours($\lambda_m = 0.001$)	28.31	0.891	0.136	2642k	21.58	0.689	0.305	4840k	26.88	0.880	0.180	7481k
Global($\lambda_m = 0.003$)	27.74	0.880	0.168	1167k	21.06	0.653	0.356	1228k	26.18	0.859	0.226	1674k
Ours($\lambda_m = 0.003$)	28.23	0.890	0.135	1901k	21.41	0.680	0.320	2742k	26.85	0.876	0.193	3714k

Table 3. **Per-scene detailed ablation study on the proposed Contribution-based Anchor Pruning.** "Base" refers to the "+AJAD" results in Tab. 1, whereas "Global" denotes the supervision strategy used in MaskGaussian [4].

Scene	Elvenruin			Road			Citysample		
Method|Metrics	**PSNR ↑**	**SSIM ↑**	**LPIPS ↓**	**PSNR ↑**	**SSIM ↑**	**LPIPS ↓**	**PSNR ↑**	**SSIM ↑**	**LPIPS ↓**
Base	28.23	0.890	0.135	21.41	0.680	0.320	26.85	0.876	0.194
w/ normal training	28.21	0.891	0.135	21.56	0.685	0.315	26.99	0.878	0.192
w/ ours	28.78	0.899	0.128	21.72	0.691	0.312	27.66	0.886	0.185

Table 4. **Per-scene detailed ablation study on the proposed Global-to-Local Optimization.** Here, "Base" denotes the model trained under the standard setting, while Normal training refers to an additional 20k iterations of training on top of "Base" using the original uniform view sampling strategy.

Scene	Elvenruin				Road				Citysample			
Method|Metrics	**PSNR ↑**	**SSIM ↑**	**LPIPS ↓**	**Anchors ↓**	**PSNR ↑**	**SSIM ↑**	**LPIPS ↓**	**Anchors ↓**	**PSNR ↑**	**SSIM ↑**	**LPIPS ↓**	**Anchors ↓**
$\gamma_{scale} = 0$	28.06	0.884	0.145	1527k	21.27	0.669	0.335	1928k	26.70	0.870	0.203	3104k
$\gamma_{scale} = 1$	28.13	0.888	0.139	1741k	21.32	0.676	0.325	2484k	26.76	0.874	0.195	3613k
$\gamma_{scale} = 2$	28.17	0.890	0.139	1837k	21.39	0.679	0.322	2622k	26.82	0.876	0.193	3742k
$\gamma_{scale} = 3$	28.23	0.890	0.135	1901k	21.41	0.680	0.320	2742k	26.85	0.876	0.193	3714k
$\gamma_{scale} = 4$	28.36	0.892	0.134	1971k	21.50	0.686	0.311	4172k	26.86	0.877	0.189	4011k

Table 5. **Effect of γ_{scale}.** To balance reconstruction quality and modeling efficiency, we adopt $\gamma_{scale} = 3$ as our final setting.

Scene	Elvenruin				Road				Citysample			
Method|Metrics	**PSNR ↑**	**SSIM ↑**	**LPIPS ↓**	**Anchors ↓**	**PSNR ↑**	**SSIM ↑**	**LPIPS ↓**	**Anchors ↓**	**PSNR ↑**	**SSIM ↑**	**LPIPS ↓**	**Anchors ↓**
Ours($\lambda_m = 0.001$)	28.31	0.891	0.136	2646k	21.58	0.689	0.305	4841k	26.88	0.880	0.180	7480k
Ours($\lambda_m = 0.002$)	28.21	0.892	0.134	2104k	21.43	0.680	0.321	2865k	26.81	0.877	0.190	4374k
Ours($\lambda_m = 0.003$)	28.23	0.890	0.135	1901k	21.41	0.680	0.320	2742k	26.85	0.876	0.193	3714k
Ours($\lambda_m = 0.004$)	28.05	0.887	0.147	1410k	21.24	0.666	0.340	1762k	26.47	0.868	0.209	3031k

Table 6. **Effect of λ_m.** To balance reconstruction quality and modeling efficiency, we adopt $\lambda_m = 0.003$ as our final setting.

Scene	Elvenruin				Road				Citysample			
Method|Metrics	**PSNR ↑**	**SSIM ↑**	**LPIPS ↓**	**Anchors ↓**	**PSNR ↑**	**SSIM ↑**	**LPIPS ↓**	**Anchors ↓**	**PSNR ↑**	**SSIM ↑**	**LPIPS ↓**	**Anchors ↓**
10k+10k	28.78	0.899	0.128	1856k	21.72	0.691	0.312	2712k	27.66	0.886	0.185	3480k
20k	28.75	0.899	0.129	1978k	21.76	0.693	0.311	2889k	27.62	0.885	0.186	3544k

Table 7. **Impact of Different Global-to-Local Optimization Designs.** The "10k+10k" setting refers to a process where uniform sampling is employed and densification is halted for the final 10k iterations. Conversely, the "20k" setting involves continuous densification for the entire 20k iterations without ever applying uniform sampling. We utilize the "10k+10k" setting for our implementation.

References

[1] Lihan Jiang, Kerui Ren, Mulin Yu, Linning Xu, Junting Dong, Tao Lu, Feng Zhao, Dahua Lin, and Bo Dai. Horizon-gs: Unified 3d gaussian splatting for large-scale aerial-to-ground scenes. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 26789–26799, 2025. 1

[2] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4), 2023. 1, 2

[3] Bernhard Kerbl, Andreas Meuleman, Georgios Kopanas, Michael Wimmer, Alexandre Lanvin, and George Drettakis. A hierarchical 3d gaussian representation for real-time rendering of very large datasets. *ACM Transactions on Graphics*, 43(4), 2024. 2

[4] Yifei Liu, Zhihang Zhong, Yifan Zhan, Sheng Xu, and Xiao Sun. Maskgaussian: Adaptive 3d gaussian representation from probabilistic masks. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 681–690, 2025. 2

[5] Zongxin Ye, Wenyu Li, Sidun Liu, Peng Qiao, and Yong Dou. Absgs: Recovering fine details in 3d gaussian splatting. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 1053–1061, 2024. 2