Confidence-Guided Multi-Scale Aggregation for Sparse-View High-Resolution 3D Gaussian Splatting

Supplementary Material

7. Extended Ablation Analysis

Hierarchical Recursion Depth (Tab. 4). We also study the effect of increasing the depth of the hierarchical recursion (the number of the gaussian fields). As the table shows, moving from two to three scales noticeably boosts performance, indicating that additional resolution levels help capture more subtle geometric cues. However, further adding a fourth or fifth scale offers diminishing returns and increases computational overhead.

Hierarchical Recursion Depth	original Res			1/2 Res		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
2	18.45	0.567	0.347	18.85	0.582	0.340
3	19.83	0.621	0.306	20.32	0.638	0.295
4	20.35	**0.631**	**0.287**	**20.91**	**0.655**	**0.258**
5	**20.37**	0.630	**0.287**	20.88	0.652	0.260

Table 4. **Ablation results of the hierarchical recursion depth.**

Confidence-Guided Aggregation (Tab. 5). We further compare our confidence-guided aggregation with a pruning-based variant that directly removes finer-scale Gaussians whose geometry deviates from their coarse anchors. While pruning can eliminate some unstable points, it also discards many partially reliable Gaussians that contribute high-frequency detail, especially under sparse-view and high-resolution settings where every observation is valuable. As shown in the table, confidence weighting consistently outperforms hard pruning at all resolutions. By softly modulating opacity instead of making binary keep-or-drop decisions, confidence-guided aggregation preserves informative but imperfect Gaussians while still suppressing noisy ones, leading to sharper details and more stable geometry.

Confidence-guided Aggregation	original Res			1/2 Res		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
×	19.23	0.613	0.315	19.87	0.636	0.299
✓	**20.35**	**0.631**	**0.287**	**20.91**	**0.655**	**0.258**

Table 5. **Ablation on the confidence-guided aggregation.**

Pseudo-View Training. Tab. 6 shows that removing pseudo-view training consistently degrades reconstruction quality. We have already experimented with intuitive joint multi-scale supervision during the method design stage, but found it to perform slightly worse than our final formulation. We attribute this to the fact that directly supervising low-resolution Gaussians tends to shift them toward fitting appearance details, which may reduce their effectiveness as structural anchors and introduce competing gradients across scales under sparse views. Our formulation benefits from treating low-resolution Gaussians primarily as anchors and enforcing cross-scale consistency via our confidence-weighted paradigm, which empirically leads to more stable optimization.

	Original Res			1/2 Res		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
ours w/o pseudo-view	18.53	0.619	0.367	19.17	0.641	0.338
joint multi-scale supervision	18.70	0.623	**0.332**	19.38	0.648	0.331
ours	**18.83**	**0.626**	0.333	**19.50**	**0.652**	**0.329**

Table 6. **Ablation of Pseudo-View Training.**

Method	Peak GPU Memory Usage (%)	Training Time (mins)
CAGS (Ours)	86%	∼35
3DGS	79%	∼30
CoR-GS	90%	∼40
FSGS	85%	∼35
DropGaussian	83%	∼30

Table 7. **Peak GPU memory usage (%) and training time (minutes).**

8. Detailed Efficiency Report

During training, we maintain most parameters consistent with those used in 3DGS. We conduct training for 10k iterations on the LLFF, DTU, Blender datasets and 30k iterations on the MipNeRF dataset, with training times ranging from 0.3 to 2 hours. Despite introducing additional multi-scale aggregation and confidence computation, our per-iteration overhead remains minimal, increasing the runtime by less than 20% compared to vanilla 3DGS. As shown in Tab. 7, the reported less than 20% overhead refers to training-time optimization compared to vanilla 3DGS, while incurring no inference-time latency, as all confidence estimation and pseudo-view mechanisms are applied only during optimization. Peak GPU memory usage increases (typically under 10%) due to maintaining auxiliary Gaussian fields at coarser scales.

9. Additional Qualitative Results

The additional qualitative results, shown in Table 8, further demonstrate the advantages of our approach. Across all datasets, our method consistently outperforms other state-of-the-art few-shot methods (FSGS [57], Binocular3DGS [15], DropGaussian [34] and NexusGS [54]) across all metrics. The improvements are particularly evident at high resolutions, where our hierarchical multi-scale framework ef-

Dataset	Resolution	Metrics	FSGS	Binocular3DGS	DropGaussian	NexusGS	Ours
Mip-NeRF360	original Res	PSNR↑	16.34	16.46	16.45	16.58	19.21
		SSIM↑	0.548	0.556	0.558	0.560	0.593
		LPIPS↓	0.276	0.272	0.263	0.271	0.234
	1/2 Res	PSNR↑	20.27	20.24	20.81	20.84	22.53
		SSIM↑	0.637	0.626	0.604	0.648	0.660
		LPIPS↓	0.312	0.324	0.239	0.230	0.204
	1/4 Res	PSNR↑	21.83	22.15	22.09	22.45	23.65
		SSIM↑	0.632	0.664	0.651	0.663	0.686
		LPIPS↓	0.231	0.183	0.203	0.174	0.145
DTU	original Res	PSNR↑	16.34	16.61	16.85	16.99	18.23
		SSIM↑	0.440	0.445	0.489	0.490	0.525
		LPIPS↓	0.478	0.481	0.466	0.459	0.420
	1/2 Res	PSNR↑	18.25	18.23	18.38	18.66	19.78
		SSIM↑	0.488	0.501	0.504	0.508	0.536
		LPIPS↓	0.480	0.442	0.464	0.433	0.404
Blender	original Res	PSNR↑	20.73	20.75	20.77	20.86	23.03
		SSIM↑	0.637	0.668	0.657	0.665	0.693
		LPIPS↓	0.331	0.312	0.287	0.281	0.235

Table 8. **Quantitative results on Mip-NeRF360 (24 views), DTU and Blender datasets.** The best, second-best, and third-best entries are marked in red, orange, and yellow, respectively. CAGS (Ours) consistently achieves superior performance, especially at higher resolutions.

	LLFF			DTU			MipNeRF-360			Blender		
	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS
3DGS	18.54	0.588	0.272	17.65	0.816	0.146	21.71	0.672	0.248	22.23	0.858	0.114
FSGS	20.43	0.682	0.248	17.14	0.818	0.162	23.40	0.733	0.238	22.70	0.822	0.137
NexusGS	21.07	0.738	0.177	20.21	0.869	0.102	23.86	0.753	0.206	24.37	0.893	0.087
DropGaussian	20.76	0.713	0.200	-	-	-	24.13	0.762	0.225	25.42	0.888	0.089
Ours	20.98	0.718	0.195	19.53	0.863	0.110	24.18	0.740	0.224	24.37	0.893	0.090

Table 9. **Quantitative results at common resolutions.** Evaluation at commonly used resolutions (downsampling rates of 8/4/4/2 for LLFF/Mip-NeRF360/DTU/Blender). Our method still achieves comparable or better results.

fectively mitigates local overfitting and noise while maintaining fine details.

Comparison at common resolutions. Tab. 9 demonstrates evaluation at commonly used resolutions (downsampling rates of 8/4/4/2 for LLFF/Mip-NeRF360/DTU/Blender). Our method achieves comparable or better results at commonly used resolutions, and consistently outperforms prior methods at higher resolutions, while preserving geometric structure and fine details.

10. Additional Visual Results

We include additional rendered views to further demonstrate the visual quality and robustness of CAGS across different scenes, as shown in Fig 6. These results illustrate that our hierarchical multi-scale framework maintains stable geometry and preserves high-frequency details.

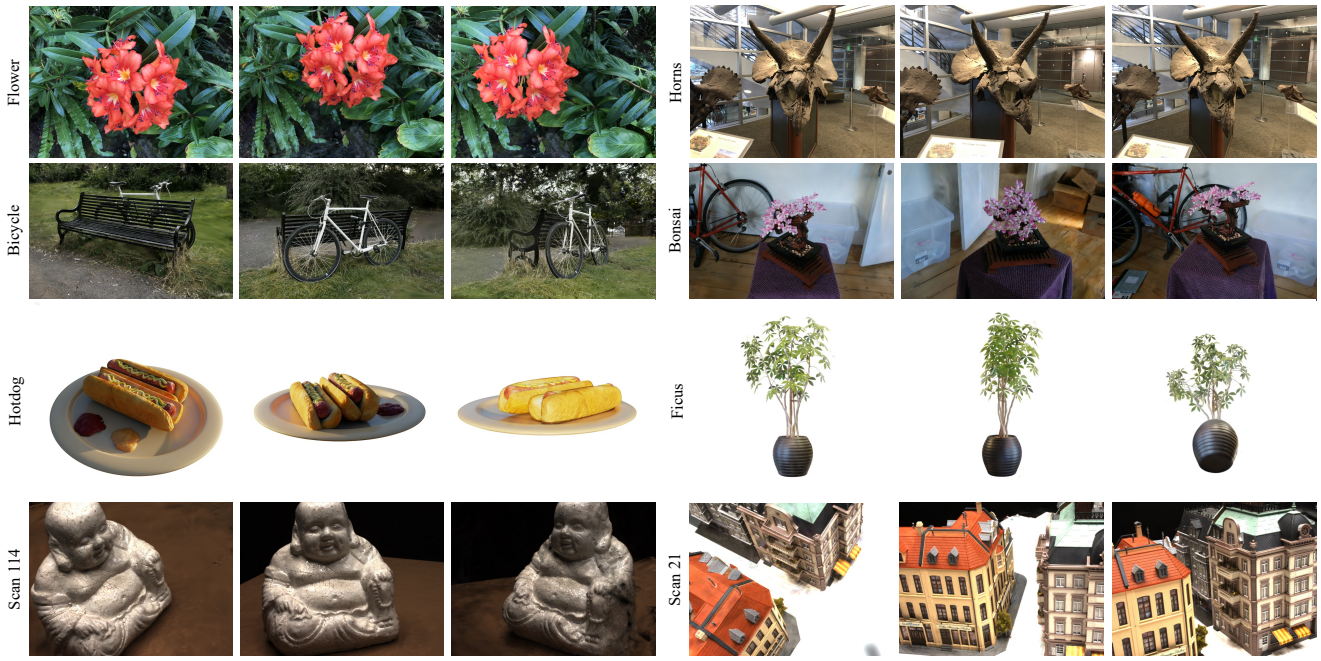

Figure 6. **Additional Visualization.** The first, second, third, and fourth rows correspond to the LLFF, Mip-NeRF360, Blender, and DTU datasets, respectively. Across all test scenes, our method consistently produces sharp edges and stable textures while significantly reducing artifacts such as floaters and ghosting. The effectiveness of our hierarchical aggregation strategy is especially evident in challenging high-resolution regions, where it successfully refines Gaussian distributions and preserves geometric coherence.