Supplementary Material: Faster 3D Gaussian Splatting Convergence via Structure-Aware Densification

ACM Reference Format:
. 2026. Supplementary Material: Faster 3D Gaussian Splatting Convergence via Structure-Aware Densification. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers (SIGGRAPH Conference Papers '26), July 19–23, 2026, Los Angeles, CA, USA.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3799902.3811212

PROPERTIES OF THE STRUCTURE ANALYSIS

Our multiscale structure-tensor analysis is built on multiscale gradients, which in their raw form encode both local contrast and frequency. The normalizations applied to the per-scale tensors (see Eq. 5 in the main paper) effectively remove amplitude effects, largely isolating the frequency component. To quantify robustness, we measure the average relative change in the aggregated tensor map $\bar{S}_{\mathrm{GT}}$ under several image perturbations.

Contrast variations. Modifying the image exposure by a factor of 0.5 or 1.5 results in an average change of only 5% or 8% in $\bar{S}_{\mathrm{GT}}$, confirming that the analysis is robust to global contrast shifts.

Noise. Due to the image pre-smoothing step (see L319 of the main paper), additive image noise likewise has little effect: adding Gaussian noise with unit standard deviation results in an average change of only 7%.

Other image degradations. As with any reconstruction algorithm, our approach is more sensitive to degradations that alter high-frequency content. Sharpening by 1.5×/3× leads to changes of 12%/40%, and JPEG compression at quality level 80 results in a change of 12%.

Semantic sensitivity. Our structure analysis responds strongly to material reflectance, specular highlights, and shadow boundaries. This behavior is desirable: these scene attributes are integral components of what a radiance field should faithfully represent, and their presence in $\bar{S}_{\mathrm{GT}}$ appropriately triggers densification in visually demanding regions.

RENDERING SPEED AND PEAK VRAM

Beyond training efficiency, we evaluate the inference-time performance of the resulting models. Table 1 reports rendering speed (frames per second, measured on an H100 GPU) and peak GPU memory consumption during training for a representative subset of methods, averaged across all datasets. Our method achieves rendering rates comparable to 3DGS [Kerbl et al. 2023] (251 vs. 296 FPS), confirming that the larger Gaussian count does not translate into a proportional rendering overhead. The higher peak VRAM relative to 3DGS [Kerbl et al. 2023] and FastGS-Big [Ren et al. 2026] reflects the additional memory required for our structure analysis and multiview consistency buffers during training, not at inference time.

Table 1. Rendering speed and peak training VRAM on an H100 GPU.

Method	FPS↑	VRAM (GB)↓
3DGS	296	9.9
FastGS-Big	415	6.6
Ours	251	13.0

COMPARISON UNDER MATCHED GAUSSIAN COUNT

Table 2 ensures a fair comparison with FastGS [Ren et al. 2026] by matching the final Gaussian count (4.1M) via an increased FastGS split factor. We evaluate this matched-capacity baseline under both its original 30k training iterations and our accelerated 3k schedule. While FastGS requires 30k iterations—and thus 10x more training time—to nearly match our quality at this Gaussian count, its performance degrades substantially when restricted to our reduced 3k iteration budget and learning rate schedule. This proves that our method's superiority stems from our fundamental formulation changes, rather than merely benefiting from a higher Gaussian count.

Table 2. Comparison under matched Gaussian count (≈4.1M) on Mip-NeRF360.

Method	Time↓	PSNR↑	SSIM↑	LPIPS↓
FastGS (30k)	502.2	27.19	0.809	0.194
FastGS (3k)	30.3	24.30	0.750	0.278
Ours (3k)	53.0	27.25	0.821	0.197

EXTENDED TRAINING

Table 3 demonstrates the rapid convergence of our method by detailing the effects of training beyond our default 3k iterations on the Mip-NeRF360 [Barron et al. 2022] dataset. While extending the training duration yields moderate overall improvements in PSNR and LPIPS, our method effectively converges very early. Notably, scene-specific analysis reveals that for outdoor scenes, there is no PSNR gap even at just 3k iterations. For indoor scenes, the remaining performance gap becomes negligible when training is extended to 30k iterations.

PROGRESSIVE ABLATION

Table 4 reports a step-by-step ablation on all scenes of the Mip-NeRF360 dataset. Starting from a base 3DGS model trained for 3k

Author's address:

SIGGRAPH Conference Papers '26, July 19–23, 2026, Los Angeles, CA, USA

Table 3. Effect of training duration on Mip-NeRF360.

Iterations	Time↓	PSNR↑	SSIM↑	LPIPS↓	N_{GS} ↓
3k (ours)	53.0	27.25	0.821	0.197	4.1M
15k	194.4	27.55	0.821	0.186	4.1M
30k	295.3	27.48	0.818	0.184	4.1M

iterations, we sequentially add our 3k LR schedule, our structure-aware densification, and our multiview consistency. Each component contributes meaningfully to the final result.

Table 4. Progressive ablation.

Method	Time↓	PSNR↑	SSIM↑	LPIPS↓	N_{GS} ↓
Base	33.2	25.84	0.748	0.296	1.5M
+ 3k LR Schedule	33.6	26.77	0.793	0.256	1.4M
+ Our Densification	54.4	27.31	0.822	0.196	4.5M
+ MV Consistency	53.0	27.31	0.821	0.197	4.0M

ABSGS DENSIFICATION

As described in the method, we apply the densification strategy of AbsGS [Ye et al. 2024] every 100 iterations to populate empty regions with primitives, ensuring that our structure-aware densification can fully leverage the existing primitive coverage. Table 5 shows that this component is not fundamental to our approach: removing it yields a slight reduction in Gaussian count and a small quality decrease (LPIPS diff of 0.005), but the method remains competitive.

Table 5. Effect of the AbsGS densification component on Mip-NeRF360.

Method	Time↓	PSNR↑	SSIM↑	LPIPS↓	N_{GS} ↓
w/ AbsGS densif.	52.96	27.25	0.821	0.197	4.05M
w/o AbsGS densif.	51.38	27.12	0.820	0.202	3.51M

EFFECT OF INITIALIZATION

Beyond the standard structure-from-motion point cloud, we additionally place Gaussians on the faces of the scene's bounding box to ensure complete geometric coverage. To assess its contribution, we run an ablation in which this initialization is omitted while keeping all other components unchanged. All quality metrics are near-identical without it (LPIPS difference of 0.002). Nevertheless, our initialization yields slight qualitative improvements in peripheral scene regions and consistently leads to a smaller total Gaussian count, as primitives are more evenly distributed from the outset.

DIFFERENCE FROM DASHGAUSSIANS

DashGaussians [Chen et al. 2025] and our method are complementary in motivation but operate on different timescales. Although both approaches leverage multi-scale reasoning, our method establishes a scale-optimal Gaussian model early in training—resolving frequency violations in the first few densification steps—whereas DashGaussians progressively refines the representation over a longer schedule. A naive combination of the two strategies would therefore not straightforwardly yield a benefit: inserting our early, proactive densification into DashGaussians' progressive pipeline disrupts its intended schedule, while appending DashGaussians' later-stage refinement after our method has already converged adds unnecessary training time without quality gain. We therefore leave a more principled integration as future work.

REFERENCES

Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. 2022. Mip-NeRF 360: Unbounded Anti-Aliased Neural Radiance Fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Youyu Chen, Junjun Jiang, Kui Jiang, Xiao Tang, Zhihao Li, Xianming Liu, and Yinyu Nie. 2025. DashGaussian: Optimizing 3D Gaussian Splatting in 200 Seconds. In *Proceedings of the Computer Vision and Pattern Recognition Conference*. 11146–11155.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 2023. 3D Gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.* 42, 4 (2023), 139–1.

Shiwei Ren, Tianci Wen, Yongchun Fang, and Biao Lu. 2026. FastGS: Training 3D Gaussian Splatting in 100 Seconds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Zongxin Ye, Wenyu Li, Sidun Liu, Peng Qiao, and Yong Dou. 2024. AbsGS: Recovering Fine Details in 3D Gaussian Splatting. In *Proceedings of the 32nd ACM International Conference on Multimedia (MM '24)*. 1053–1061. https://doi.org/10.1145/3664647.3681361