SAGS: Structure-Aware 3D Gaussian Splatting Supplementary Material

Evangelos Ververas[1,2*], Rolandos Alexandros Potamias[1,2*], Jifei Song[2], Jiankang Deng[1,2], and Stefanos Zafeiriou[1]

[1] Imperial College London, UK
[2] Huawei Noah's Ark Lab, UK
https://ververas.github.io/SAGS/
* Equal contribution

1 Implementation Details.

We build our model on top of the original 3D-GS [4] PyTorch implementation. Similar to 3D-GS, we train our method for 30,000 iterations across all scenes and apply a growing and pruning step until 15,000 iterations, every 100 iterations, starting from the 1500 iterations. Throughout our implementation, we utilized small MLPs with a hidden size of 32. Our hash-grid positional encoder had 16 levels, with 2 features per level and a max size of hash maps 2^{19}. We used Tanh as the final activation function for the opacity MLP and Sigmoid for the color MLP while for the covariance MLP we opted not to use any activation function. To estimate the curvature of each point cloud we used k=10 neighbours [8], while for the densification step we selected k=10 neighbours for the low-curvature and k=2 neighbours for the high-curvature regions. We trained our model using Adam optimizer. We set the learning rate of opacity MLP to 0.002, of color to 0.008, of the displacements GNN to 0.01 and to the covariance MLP to 0.004. All learning rates were subject to exponential scheduling. We set the minimum opacity threshold to 0.005. In our SAGS-Lite implementation we set k=5 midpoints.

2 Extensive Results.

Per-Scene Results. In this section we report extensive per-scene results for the novel-view synthesis experiments presented in the main paper. In particular, in Tab. 1 we evaluate the rendering quality of each scene in Tanks&Temples [5] and Deep Blending [3] scenes and Tab. 2 on the MipNeRF-360 [1] dataset. In Tab. 3, we compare the proposed SAGS and SAGS-Lite models with the 3D-GS and Scaffold-GS models in terms of per-scene size. As can be easily seen the proposed models can outperform the baseline methods in terms of visual quality while, at the same, reducing the storage requirements of the model.
Multi-Scale Scenes. In addition to the aforementioned datasets, we also evaluated the performance of the proposed model in more challenging scenes using the multi-scale BungeeNeRF dataset [9]. As can be seen in Tab. 4, the proposed

Table 1: Per-scene rendering quality evaluation for Tanks&Temples [5] and Deep Blending [3] datasets.

Scene	Truck			Train			Dr Johnson			Playroom		
Method	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS
3D-GS [4]	25.19	0.879	0.148	21.10	0.802	0.218	28.77	0.899	0.244	30.04	0.906	0.241
Mip-NeRF360 [1]	24.91	0.857	0.159	19.52	0.660	0.354	29.14	0.901	0.237	29.66	0.900	0.252
iNPG [7]	23.26	0.779	0.274	20.17	0.666	0.386	27.75	0.839	0.381	19.48	0.754	0.465
Plenoxels [2]	23.22	0.774	0.335	18.93	0.663	0.422	23.14	0.787	0.521	22.98	0.802	0.499
Scaffold-GS [6]	25.77	0.883	0.147	22.15	0.822	0.206	29.80	0.907	0.250	30.62	0.904	0.258
Ours-Lite	25.95	0.881	0.143	22.36	0.810	0.220	28.61	0.890	0.285	29.53	0.888	0.300
Ours	26.31	0.894	0.134	23.44	0.837	0.198	29.99	0.912	0.234	30.96	0.913	0.248

Table 2: Per-scene rendering quality evaluation for Mip-NeRF360 [1] dataset.

Scene	bicycle			garden			stump			room			counter			kitchen			bonsai		
Method	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS
3D-GS [4]	25.25	0.771	0.205	27.41	0.868	0.103	26.55	0.775	0.210	30.63	0.914	0.220	28.70	0.905	0.204	30.32	0.922	0.129	31.98	0.938	0.205
MNeRF360 [1]	24.37	0.685	0.301	26.98	0.813	0.170	26.40	0.744	0.261	31.63	0.913	0.211	29.55	0.894	0.204	32.23	0.920	0.127	33.46	0.941	0.176
iNPG [7]	22.19	0.491	0.487	24.60	0.649	0.312	23.63	0.574	0.450	29.27	0.855	0.301	26.44	0.798	0.342	28.55	0.818	0.254	30.34	0.890	0.227
Plenoxels [2]	21.91	0.496	0.506	23.49	0.606	0.386	20.66	0.523	0.503	27.59	0.841	0.419	23.62	0.759	0.441	23.42	0.648	0.447	24.67	0.814	0.398
Scaffold-GS	24.50	0.705	0.306	27.17	0.842	0.146	26.27	0.784	0.284	31.93	0.925	0.202	29.34	0.914	0.191	31.30	0.928	0.126	32.70	0.946	0.185
Ours-Lite	24.41	0.719	0.287	25.62	0.775	0.230	25.17	0.706	0.317	31.67	0.923	0.203	28.93	0.910	0.194	30.70	0.921	0.137	33.31	0.933	0.210
Ours	25.30	0.759	0.231	27.25	0.837	0.151	26.87	0.776	0.230	32.27	0.933	0.183	29.86	0.922	0.176	31.96	0.935	0.115	34.04	0.955	0.167

Table 3: Storage size (MB) comparison between the proposed and the baseline methods. The proposed method achieves high fidelity rendering quality while effectively reducing the storage requirements of the scene.

Method | Scenes	Truck	Train	Dr Johnson	Playroom	bicycle	garden	stump	room	counter	kitchen	bonsai
3D-GS [4]	578	240	715	515	1291	1268	1034	327	261	414	281
Scaffold-GS [6]	107	66	69	63	248	271	493	133	194	173	258
SAGS	89	61	63	53	195	128	199	104	95	99	134
SAGS-Lite	40	29	32	24	113	76	103	52	42	43	77

method can outperform Scaffold-GS and 3D-GS baseline while effectively reducing the size of the scene by up to 12.4 × using our full model and 36.7 × using SAGS-Lite. Importantly, SAGS-Lite demonstrates superior performance compared to the 3D-GS method across all scenes, often by a substantial margin, while also requiring significantly less memory. Qualitative results on rendering quality and structure preservation are provided in Fig. 1 and Fig. 2 respectively.

Efficacy of neighborhood size. We further investigated the performance of the proposed model under different neighborhood sizes k. In particular, in Tab. 5 we report the performance of SAGS, while using a different number of neighbours k for the densification step and the aggregation function of the GNN. We observed that k=10 neighbors achieved the best performance while retaining the compact size of the model. As can be easily seen using larger values of k for the densification step, despite achieving similar rendering quality, results in larger

Table 4: PSNR (dB) and Storage size (MB) for BungeeNeRF [9] scenes.

Scene	Amsterdam		Bilbao		Pompidou		Quebec		Rome		Hollywood	
Method	PSNR	Mem	PSNR	Mem	PSNR	Mem	PSNR	Mem	PSNR	Mem	PSNR	Mem
3D-GS [4]	25.74	1453	26.35	1337	21.20	2129	28.79	1438	23.54	1626	23.25	1642
Scaffold-GS	27.10	243	27.66	197	25.34	230	30.51	166	26.50	200	24.97	182
Ours-Lite	25.89	68	27.12	55	24.73	58	28.76	55	25.36	64	24.21	58
Ours	27.48	228	27.91	150	25.54	177	30.72	113	26.57	153	25.21	132

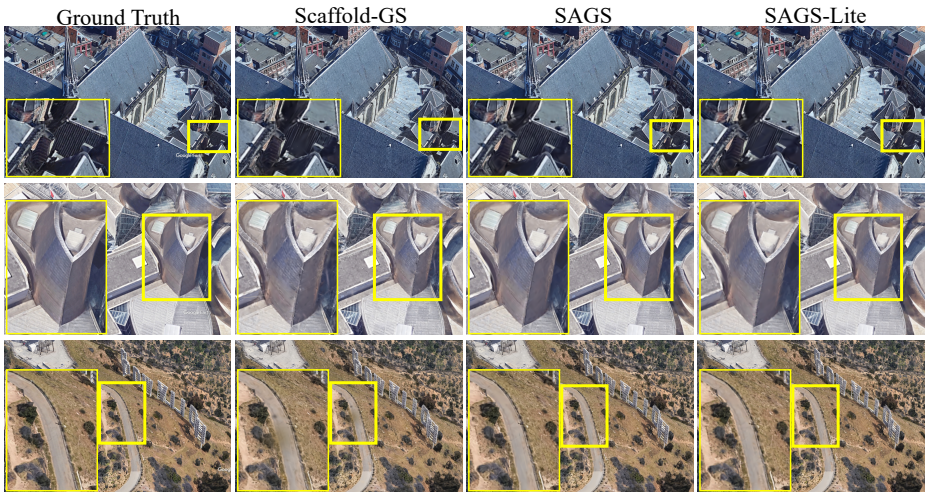

Fig. 1: Depth Structural Preservation. Comparison between the proposed and the Scaffold-GS method on the BungeeNeRF dataset. The proposed method can accurately capture sharp edges and high frequency details that Scaffold-GS method fails to model.

models. On the contrary, using smaller neighborhoods, although reducing the storage requirements, results in lower quality renderings.

Similarly, by aggregating large neighbourhoods on the GNN reflects in a rendering quality drop since each Gaussian becomes depended to distant Gaussians with dissimilar attributes. Using smaller neighborhoods, translates in larger storage requirements since more points are generated at every densification step. We opted to select k=10 since it was a sweet spot between model size and rendering performance.

3 Limitations and Ethical Considerations.

3D-GS has two severe limitations: 1) As has been extensively highlighted in the literature [4, 6], SfM techniques often struggle to generate 3D points in textureless regions, leading to empty areas. Consequently, the densification strategy

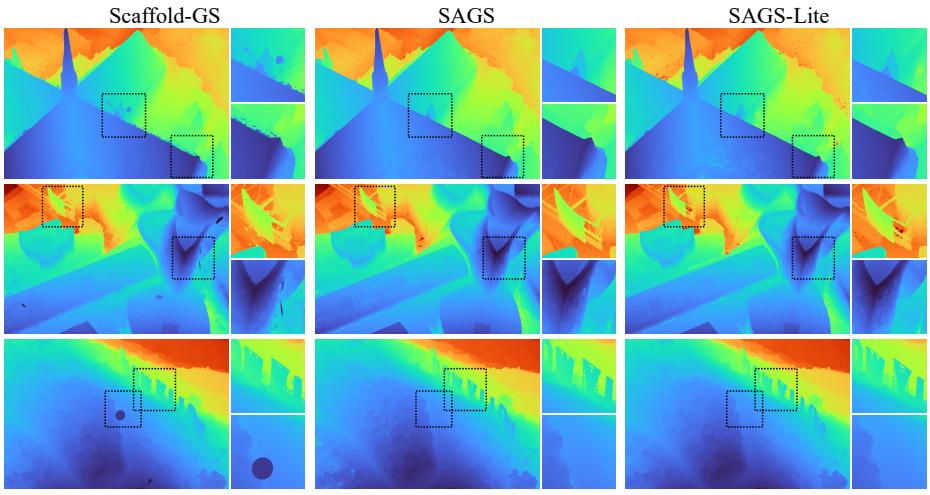

Fig. 2: Depth Structural Preservation. Comparison between the depth maps of the proposed and the Scaffold-GS method on the BungeeNeRF dataset. As can be seen from the depth maps, both SAGS and SAGS-Lite models outperform the structural preservation of Scaffold-GS. Note that the proposed models can not only preserve the sharp edges of the scene but also suppress "floater" artifacts that are visible on the Scaffold-GS depth maps.

Table 5: Ablation study on the number of neighbours k. We evaluated the influence of number of neighbours selected in densification and in the aggregation function. The reported experimentation of various k values was performed on the Deep Blending and the Tanks&Temples datasets.

Scene	Deep Blending				Tanks&Temples			
Ablation | Metrics	PSNR ↑	SSIM ↑	LPIPS ↓	Mem ↓	PSNR ↑	SSIM ↑	LPIPS ↓	Mem ↓
Densification k= 2	29.89	0.903	0.256	46	24.00	0.853	0.173	60
Densification k= 5	30.11	0.908	0.248	50	24.48	0.859	0.169	68
Densification k= 10	**30.47**	**0.913**	**0.241**	58	24.88	**0.866**	**0.166**	75
Densification k= 15	30.46	0.912	0.243	66	**24.89**	0.865	0.167	88
GNN k= 5	30.17	0.910	0.247	68	24.38	0.852	0.160	82
GNN k= 10	**30.47**	**0.913**	**0.241**	58	**24.88**	**0.866**	**0.166**	75
GNN k= 15	30.42	0.911	0.244	59	24.85	0.863	0.169	75
GNN k= 20	30.01	0.909	0.250	61	24.20	0.847	0.178	73

faces challenges in creating reliable Gaussians to encompass the scene due to inadequate initialization. 2) Similar to the initialization of 3D-GS, the growing operator builds upon the existing Gaussians which could further limit the generation of new Gaussians in under-represented areas. We attempted to tackle both such limitations using an initial densification step that populates the low-curvature regions that are usually under-sampled. However, relying on k-NN

estimates (i.e. curvature and normals) for identifying these regions could not ensure the mitigation of such issues.

Generating novel views, especially when training data is sparse, may result in inaccurate scene details. This is particularly important in real-world applications such as GPS maps, where it's crucial to understand that these methods provide estimates.

References

1. Barron, J.T., Mildenhall, B., Verbin, D., Srinivasan, P.P., Hedman, P.: Mip-nerf 360: Unbounded anti-aliased neural radiance fields. CVPR (2022)
2. Fridovich-Keil, S., Yu, A., Tancik, M., Chen, Q., Recht, B., Kanazawa, A.: Plenoxels: Radiance fields without neural networks. In: CVPR (2022)
3. Hedman, P., Philip, J., Price, T., Frahm, J.M., Drettakis, G., Brostow, G.: Deep blending for free-viewpoint image-based rendering. ACM Transactions on Graphics (ToG) **37**(6), 1–15 (2018)
4. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G.: 3d gaussian splatting for real-time radiance field rendering. ACM Transactions on Graphics **42**(4) (July 2023), https://repo-sam.inria.fr/fungraph/3d-gaussian-splatting/
5. Knapitsch, A., Park, J., Zhou, Q.Y., Koltun, V.: Tanks and temples: Benchmarking large-scale scene reconstruction. ACM Transactions on Graphics **36**(4) (2017)
6. Lu, T., Yu, M., Xu, L., Xiangli, Y., Wang, L., Lin, D., Dai, B.: Scaffold-gs: Structured 3d gaussians for view-adaptive rendering. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2024)
7. Müller, T., Evans, A., Schied, C., Keller, A.: Instant neural graphics primitives with a multiresolution hash encoding. ACM Transactions on Graphics (ToG) **41**(4), 1–15 (2022)
8. Potamias, R.A., Neofytou, A., Bintsi, K.M., Zafeiriou, S.: Graphwalks: efficient shape agnostic geodesic shortest path estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2968–2977 (2022)
9. Xiangli, Y., Xu, L., Pan, X., Zhao, N., Rao, A., Theobalt, C., Dai, B., Lin, D.: Bungeenerf: Progressive neural radiance field for extreme multi-scale scene rendering. In: Avidan, S., Brostow, G.J., Cissé, M., Farinella, G.M., Hassner, T. (eds.) Computer Vision - ECCV 2022 - 17th European Conference, Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part XXXII. Lecture Notes in Computer Science, vol. 13692, pp. 106–122. Springer (2022). https://doi.org/10.1007/978-3-031-19824-3_7, https://doi.org/10.1007/978-3-031-19824-3_7