Surface Reconstruction from 3D Gaussian Splatting via Local Structural Hints
Supplementary Material

Qianyi Wu[1], Jianmin Zheng[2], and Jianfei Cai[1,2]

[1] Department of Data Science and AI, Monash University
[2] College of Computing and Data Science, Nanyang Technological University
{qianyi.wu,jianfei.cai}@monash.edu, asjmzheng@ntu.edu.sg

This document includes the supplementary information of our main document. We provide more details about our experiments in Sec. A, including implementation details, metric design, and dataset acquisition. In the following section, we provide more experimental results on Replica and Scannet in Sec. B and Sec. C respectively. Finally, we discuss the limitation of our framework in Sec. D.

A More details about experiments

We provide more details about the experiment, including the dataset preprocess, implementation details, and evaluation metric definition.

A.1 More Implementation details

Our implementation is based on Scaffold-GS [5]. Following [5], we set an opacity MLP, covariance MLP, and color MLP to decode the corresponding attributes for each spawn Gaussian. We set the anchor feature size as 32. As the main focus of our framework targets surface reconstruction, we take the anchor feat as input for opacity MLP and covariance MLP to get the opacity, covariance, and scaling for each spawns Gaussian, while using the concatenated feature by anchor feature and view direction encoding for the color MLP to get the view-depend appearance, which is different with original design. We set the neural implicit network following the structure of Instant-NGP [6]. For the experiments conducted in Replica, we used the Poisson octree depth as 8 for surface extraction. We adopt the default train/test split for the Replica dataset. For ScanNet, we follow the experimental design of MonoSDF [12] which uses the entire dataset for training and evaluating the surface quality. We set the voxel size to 0.001 for the initial anchor point construction. We provide the training overview in Fig. A.

We provide more details about the point sampling strategy used in our framework for the joint optimization of neural implicit representation. Firstly, we use the depth rendered by the Gaussians to un-project a 2D pixel into a 3D point. To guarantee the effectiveness of the point in the calculation of MLS, we propose to find the nearest Gaussian for this point and use the corresponding Gaussian

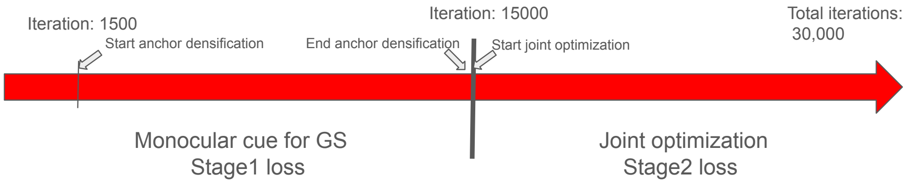

Fig. A: The training progress of our framework. We follow the Scaffold-GS to start anchor densification in 1500 iterations. Once the anchors stop growing or pruning, we add the neural implicit network for joint optimization to regulate the Gaussian attributes till the end of training.

to generate a new candidate with its mean and covariance. We will use this new point and its nearest L Gaussians to calculate the MLS function. As suggested in [4,10], we use the sampling points to construct a sphere with a radius of 0.01 and only use these Gaussians inside the sphere for the MLS function calculation.

A.2 The dataset for training

We use Replica and ScanNet for experiments. We use the ground camera poses for both datasets to train our model. To obtain the sparse point cloud used for 3D Gaussian Splatting, we use those ground truth camera poses and rerun COLMAP [8] to construct the initial point cloud. We also trained SuGaR with the ground truth camera pose and the initialized sparse point cloud for comparison.

We also normalized the camera pose to make sure the scene is located in a unit cube for the construction of instant-ngp [6] hash embedding. While this can also be implemented dynamically using the anchor point to set the normalized factor.

A.3 Metric for evaluating surface reconstruction

The definition of the evaluation metrics we used in the main document is given in Table. A.

A.4 Details about RIMLS

The detailed equation of RIMLS is defined as follows:

$$F_{\mathrm{RIMLS}}(\boldsymbol{x}) = \frac{\sum_{l=1}^{L} o_l G_l(\boldsymbol{x}) \phi(\|\boldsymbol{n}_l - \frac{\nabla F_{\mathrm{MLP}}(\boldsymbol{x})}{\|\nabla F_{\mathrm{MLP}}(\boldsymbol{x})\|}\|) \langle \boldsymbol{x} - \boldsymbol{\mu}_l, \boldsymbol{n}_l \rangle}{\sum_{l=1}^{L} o_l G_l(x) \phi(\|\boldsymbol{n}_l - \frac{\nabla F_{\mathrm{MLP}}(\boldsymbol{x})}{\|\nabla F_{\mathrm{MLP}}(\boldsymbol{x})\|}\|)}. \quad (1)$$

As mentioned in the main document, the ϕ is defined as a 1-D Gaussian kernel with a variance of σ_n^2. The σ_n is set as 0.05 in our implementation.

Metric	Definition
Accuracy	$\text{mean}_{\mathbf{p}\in\mathbf{P}}(\min_{\mathbf{q}\in\mathbf{Q}} \|\mathbf{p}-\mathbf{q}\|_1)$
Completeness	$\text{mean}_{\mathbf{q}\in\mathbf{Q}}(\min_{\mathbf{p}\in\mathbf{P}} \|\mathbf{p}-\mathbf{q}\|_1)$
Chamfer-L1	0.5 * (Accuracy + Completeness)
Precision	$\text{mean}_{\mathbf{p}\in\mathbf{P}}(\min_{\mathbf{q}\in\mathbf{Q}} \|\mathbf{p}-\mathbf{q}\|_1) < 0.05$
Recall	$\text{mean}_{\mathbf{q}\in\mathbf{Q}}(\min_{\mathbf{p}\in\mathbf{P}} \|\mathbf{p}-\mathbf{q}\|_1) < 0.05$
F-score	$2 * \text{Precision} * \text{Recall}/(\text{Precision}+\text{Recall})$

Table A: Evaluation Metric Calculation. We provide the equation for computing the quantitative metric used in the experiment. Given the sampled point cloud from ground-truth $\mathbf{P}$ and predicted result $\mathbf{Q}$, all the metrics can be calculated as shown above.

B More results on Replica

To further investigate the benefits of joint optimization, we conducted additional experiments focused on using an implicit network to fit normal-augmented Gaussians.

Why need joint optimization? For this purpose, we targeted Gaussians derived from Scaffold+N&D [5] and employed an identical implicit network structure for fitting. The training process utilized the same loss functions as in our joint optimization scheme, $\mathcal{L}_{\texttt{joint}}$. The critical variable in our experiment was the application of joint optimization for updating the Gaussians. We visualized the resulting meshes by the implicit network through Marching Cubes, as depicted in Fig. B. Our findings reveal that without joint optimization, the implicit function tends to fit more high-frequency noise as the number of iterations increases, leading to a less smooth surface. Conversely, the use of joint optimization yields a smoother, more accurate surface by regulating the Gaussians' positions and orientations. This regulation not only mitigates the tendency to fit low-frequency details early on but also refines the Gaussians' attributes for better surface alignment. Notably, meshes generated through our method exhibited some floating elements in empty spaces, attributed to isolated Gaussians. However, by employing Poisson reconstruction for surface construction from the final Gaussians, our approach demonstrates robustness against such noisy and outlier Gaussians, ensuring cleaner, more coherent mesh outputs.

Visual quality comparison with SuGaR. While our focus is on surface reconstruction, as requested we compare our view synthesis results with the refinement output from SuGaR (SDF). The average results over 8 Replica scenes are reported below (left) in Tab. B. Our method achieves superior visual quality in PSNR and SSIM. This is mainly because of the implicit appearance modeling inherited from Scaffold-GS. Besides better visual quality, our method achieves much better surface reconstruction results compared with SuGaR.

Ablation study of octree depth. Table B (right) also presents surface reconstruction results across different octree depths, adhering to SuGaR's recommended guidelines to ensure fairness, specifically using an octree depth of 8 as outlined in line 25 of the supplementary material. While increasing the octree

Task&	Visual Quality	Surface Quality (F-score/	Chamfer Distance)	
Setting	PSNR/SSIM/LPIPS	depth=8	depth=9	depth=10
SuGaR	37.07/0.9659/**0.0593**	52.10/9.57	53.55/9.16	54.59/9.08
Ours	**37.53**/**0.9704**/0.0856	**67.22/7.08**	**67.79/6.97**	**67.93/6.94**

Table B: Comparision with SuGaR on visual quality (left) and surface reconstruction under different octree-depth (right).

depth leads to improved quantitative results, it also results in a rougher surface. Across all tested octree depths, our method consistently outperforms SuGaR.

MLS SDF *vs* SuGaR SDF. We replaced the SDF estimation method in our framework with the SuGaR SDF design while keeping all other components unchanged, and evaluated it across 8 Replica scenes. As shown below, our configuration with SuGaR SDF underperforms compared to our original method in all metrics. This reinforces the superiority of our MLS SDF, which provides more precise information that significantly enhances geometric detail.

Setting	Normal-C ↑	F-score ↑	CD ↓
Ours	**85.23**	**67.22**	**7.08**
Ours w/ SuGaR SDF estimation	83.89	65.59	7.68
SuGaR w/ monocular guidance	77.82	58.42	8.71
SuGaR	76.11	52.10	9.57

Table C: SuGaR with our proposed designs.

SuGaR with monocular guidance. The table above C also reports the results of incorporating monocular guidance with SuGaR, which improves all geometry quality metrics over vanilla SuGaR, yet they still fall short of ours. Introducing monocular cues at the initial GS stage of SuGaR offers effective initialization. However, it results in blocky artifacts and inferior performance compared to ours due to SuGaR's regularization.

C More results on ScanNet

We provide more comprehensive results on ScanNet [1] in Tab. D. Our approach achieves a comparable performance with Manhattan-SDF [3] and MonoSDF (Grids) [12] in all metrics, which demonstrates a strong potential ability of 3DGS to produce a high-quality surface mesh.

D Discussion about Limitations

While our method achieves superior quality in many aspects, our analysis reveals a noticeable gap when compared to the current state-of-the-art in surface reconstruction. A promising direction for future work is the incorporation of appearance guidance, which can enable more precise capture of detailed geometry, thereby lessening our method's dependence on pre-trained models. Furthermore,

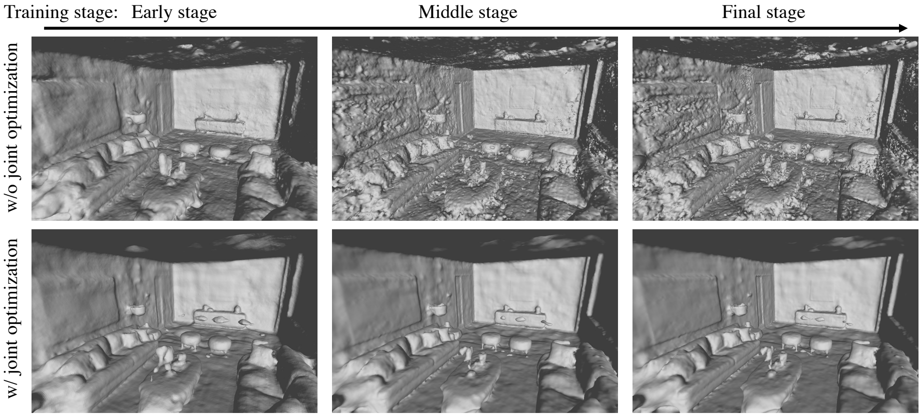

Fig. B: Comparison of the mesh produced by performing Marching Cube from the implicit network. The top row shows the mesh produced by the neural implicit network without joint optimization of the Gaussians and the bottom row depicts the mesh from the implicit network with joint optimization. We noticed that without joint optimization, the implicit function will fit the high-frequency noise as the training goes on. The joint optimization regulates the orientation and position of Gaussians to obtain a better surface alignment Gaussian Splatting field.

revisiting and potentially revising the Gaussian assumption inherent in our 3D Gaussian Splatting (3DGS) approach could yield more accurate models for density estimation, enhancing surface reconstruction fidelity. These areas represent valuable opportunities for future research, signaling our commitment to pushing the boundaries of what is achievable in surface reconstruction quality.

	Accuracy ↓	Completeness ↓	Chamfer-L1 ↓	Precision ↑	Recall ↑	F-Score ↑
UNISURF [7]	0.554	0.164	0.359	0.212	0.362	0.267
Neus [9]	0.179	0.208	0.194	0.313	0.275	0.291
VolSDF [11]	0.414	0.120	0.267	0.321	0.394	0.346
Manhattan-SDF [3]	0.072	0.068	0.070	0.621	0.586	0.602
MonoSDF (Grids) [12]	0.072	0.057	0.064	0.660	0.601	0.626
MonoSDF (MLP) [12]	0.035	0.048	0.042	0.799	0.681	0.733
SuGaR [2](density)	0.398	0.170	0.284	0.189	0.255	0.217
SuGaR [2](SDF)	0.328	0.211	0.269	0.179	0.189	0.184
Ours	0.067	0.069	0.068	0.604	0.594	0.599

Table D: The quantitative results of the scene reconstruction on ScanNet.

References

1. Dai, A., Chang, A.X., Savva, M., Halber, M., Funkhouser, T., Nießner, M.: ScanNet: Richly-annotated 3d reconstructions of indoor scenes. In: CVPR (2017) 4
2. Guédon, A., Lepetit, V.: Sugar: Surface-aligned gaussian splatting for efficient 3d mesh reconstruction and high-quality mesh rendering. In: CVPR (2024) 5
3. Guo, H., Peng, S., Lin, H., Wang, Q., Zhang, G., Bao, H., Zhou, X.: Neural 3d scene reconstruction with the manhattan-world assumption. In: CVPR (2022) 4, 5
4. Liu, S.L., Guo, H.X., Pan, H., Wang, P.S., Tong, X., Liu, Y.: Deep implicit moving least-squares functions for 3d reconstruction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1788–1797 (2021) 2
5. Lu, T., Yu, M., Xu, L., Xiangli, Y., Wang, L., Lin, D., Dai, B.: Scaffold-gs: Structured 3d gaussians for view-adaptive rendering. In: CVPR (2024) 1, 3
6. Müller, T., Evans, A., Schied, C., Keller, A.: Instant neural graphics primitives with a multiresolution hash encoding. ACM TOG. **41**(4), 102:1–102:15 (Jul 2022) 1, 2
7. Oechsle, M., Peng, S., Geiger, A.: Unisurf: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction. In: ICCV (2021) 5
8. Schonberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: CVPR. pp. 4104–4113 (2016) 2
9. Wang, P., Liu, L., Liu, Y., Theobalt, C., Komura, T., Wang, W.: Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. In: NeurIPS (2021) 5
10. Wang, Z., Wang, P., Wang, P., Dong, Q., Gao, J., Chen, S., Xin, S., Tu, C., Wang, W.: Neural-imls: Self-supervised implicit moving least-squares network for surface reconstruction. IEEE Transactions on Visualization and Computer Graphics pp. 1–16 (2023). `https://doi.org/10.1109/TVCG.2023.3284233` 2
11. Yariv, L., Gu, J., Kasten, Y., Lipman, Y.: Volume rendering of neural implicit surfaces. In: NeurIPS (2021) 5
12. Yu, Z., Peng, S., Niemeyer, M., Sattler, T., Geiger, A.: Monosdf: Exploring monocular geometric cues for neural implicit surface reconstruction. In: NeurIPS (2022) 1, 4, 5