Supplementary Material of "QuickSplat: Fast 3D Surface Reconstruction via Learned Gaussian Initialization"

1. More Implementation Details

SuGaR. We follow the official code that optimizes vanilla 3DGS for 7,000 iterations and refine for 15,000 iterations to get the best quality mesh. Depth-normal consistency (dn_consistency) is used as the regularization objective.

2DGS. We follow the official code and optimize the scene for 30,000 iterations, using the same hyper-parameters such as the learning rates and the number of iterations for pruning and densification; we only optimize the RGB color of the Gaussians instead of the spherical harmonics.

GS2Mesh. We follow the official code and optimize vanilla 3DGS for 30,000 iterations. The pretrained stereo estimation model from DLNR [6] that is trained on Middlebury is used to extract stereo depth, with 0.1m as the stereo baseline. Since we work on scene-level datasets, the object masks are ignored.

MonoSDF. We follow the official code and use MLP as the scene representation. We use the Omnidata [3] to extract the depth and normal of the training images, and both depth and normal losses are used for the optimization. The model is optimized for 1,000 epochs.

PGSR. (Chen et al. 2024) We uses the official code and optimize the scenes for 30,000 iterations, with single view and multi-view regularization loss after 7,000 iterations. Exposure compensation is not used as ScanNet++ has fixed camera exposure.

QuickSplat. We provide the pseudo code of the optimization process of QuickSplat in Algorithm 1.

Algorithm 1 The optimization process of QuickSplat

$\mathcal{P}$: SfM points
f_I: initializer network
f_D: densifier network
f_O: optimizer network

$\mathcal{G}_0 \leftarrow f_I(\mathcal{P})$
for $t = 0$ to $T-1$ **do**
  $\nabla\mathcal{G}_t \leftarrow 0$
  for all images **do**
    $L \leftarrow$ rendering loss of the image
    $\nabla\mathcal{G}_t \leftarrow \nabla\mathcal{G}_t + \frac{\delta L}{\delta \mathcal{G}_t}$
  end for
  $\hat{\mathcal{G}}_t \leftarrow f_D(\mathcal{G}_t, \nabla\mathcal{G}_t, t)$
  $\bar{\mathcal{G}}_t \leftarrow \mathcal{G}_t \cup \hat{\mathcal{G}}_t$ ▷ Concatenate the new GS

  $\nabla\bar{\mathcal{G}}_t \leftarrow \nabla\mathcal{G}_t \cup \mathbf{0}$
  $\Delta\bar{\mathcal{G}}_t \leftarrow f_O(\bar{\mathcal{G}}_t, \nabla\bar{\mathcal{G}}_t, t)$
  $\mathcal{G}_{t+1} \leftarrow \bar{\mathcal{G}}_t + \Delta\bar{\mathcal{G}}_t$ ▷ Update the parameters
end for

2. Additional results

Generalization. To demonstrate the generalization ability of our method, we run QuickSplat trained on ScanNet++ directly on other indoor datasets, such as ARKitScenes [2] and Mip-NeRF 360 [1], without any additional fine-tuning.

We process the ARKitScenes dataset following the same procedure as ScanNet++, obtaining the SfM point clouds and the alignment between camera poses and the ground-truth mesh. For Mip-NeRF 360 (Room), we restore the absolute scale of the official COLMAP point cloud and poses using a monocular metric depth estimator [5].

This cross-dataset setting is more challenging due to the domain gap between datasets. Additionally, the RGB captures in ARKitScenes and Mip-NeRF 360 have a smaller field of view compared to ScanNet++, making reconstruction from images more difficult. We compare QuickSplat with 2DGS in Tab. 1 and Fig. 1, which demonstrate the generalization capability of our proposed method. Additional reconstruction results are shown in Fig. 2.

Method	Abs err↓	Acc (10cm)↑	Chamfer↓	Time
2DGS	0.6978	0.3590	0.6015	1780s
Ours	**0.1775**	**0.7698**	**0.4301**	111s

Table 1. **Evaluation on ARKitScenes (5 scenes, no fine-tuning).**

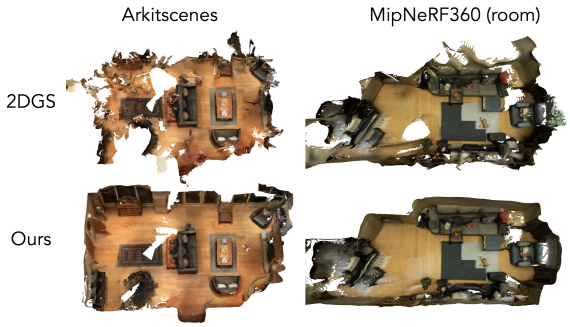

Figure 1. **Ours *vs*. 2DGS on ARKitScenes and MipNeRF 360.** To demonstrate the generalization ability of QuickSplat, we run our model on ARKitScenes [2] and Mip-NeRF 360 [1] without fine-tuning. Compared to 2DGS, QuickSplat produces more complete geometry

Large scenes. We also demonstrate the capability to reconstruct larger scenes (*e.g.*, indoor scenes containing multiple rooms) in Fig. 3, as the method is not constrained by the number of input images. Note that the optimization times for larger scenes would increase due to the increasing number of frames during gradient accumulation. However, the overall time is still substantially faster than the existing methods.

3. Additional ablations

Steps We ablate the number of steps for the learned optimizer and post-optimization in Tab. 2. We observe that the depth error decreases gradually over the 5 optimization steps. Additional SGD optimization steps lead to a plateau and require more time. On the other hand, the Chamfer distance changes only marginally due to the good global geometry generated by our learned initialization.

	$T=0$	$T=1$	$T=2$	$T=5$	SGD=1k	SGD=2k
Abs err↓	0.0921	0.0881	0.0807	0.0732	0.0598	0.0578
Rel err↓	0.0923	0.0792	0.0568	0.0431	0.0314	0.0292
Chamfer↓	0.1478	0.1437	0.1448	0.1461	0.1361	0.1347
Time (s)	0.6	5.7	11	26	77	124

Table 2. **Ablation over time steps.**

Optimization and densification We experiment with combining QuickSplat initialization with the original 2DGS optimization and densification, instead of using our optimization and densification networks, under comparable time constraints. As shown in Tab. 3, the learnable optimization and densification networks achieve better reconstruction in finer details (i.e., the accuracy metrics with small thresholds). Although the original SGD optimization and densification benefit from our initialization, our full method remains more efficient.

Extend initializer to other method We demonstrate that our initializer can be easily integrated into other Gaussian splatting variants, such as SAGS [4]. Note that we modified SAGS to use 2D Gaussians instead of 3D Gaussians as the representation for reconstructing 3D surfaces. As shown in Tab. 4, SAGS with our initialization performs significantly better than with SfM initialization. Moreover, our full method, with the learned optimization and densification, reconstructs scenes more accurately and efficiently than SAGS's original optimization and densification.

References

[1] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5470–5479, 2022. 1, 2

[2] Gilad Baruch, Zhuoyuan Chen, Afshin Dehghan, Tal Dimry, Yuri Feigin, Peter Fu, Thomas Gebauer, Brandon Joffe, Daniel Kurz, Arik Schwartz, and Elad Shulman. ARKitscenes - a diverse real-world dataset for 3d indoor scene understanding using mobile RGB-d data. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*, 2021. 1, 2

[3] Ainaz Eftekhar, Alexander Sax, Jitendra Malik, and Amir Zamir. Omnidata: A scalable pipeline for making multi-task mid-level vision datasets from 3d scans. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10786–10796, 2021. 1

[4] Evangelos Ververas, Rolandos Alexandros Potamias, Jifei Song, Jiankang Deng, and Stefanos Zafeiriou. Sags: Structure-aware 3d gaussian splatting. *arXiv:2404.19149*, 2024. 2, 3

[5] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. *arXiv:2406.09414*, 2024. 1

[6] Haoliang Zhao, Huizhou Zhou, Yongjun Zhang, Jie Chen, Yitong Yang, and Yong Zhao. High-frequency stereo matching network. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1327–1336, 2023. 1

Figure 2. **More reconstruction result of QuickSplat on ARKitscenes dataset.**

Figure 3. **Additional qualitative results of QuickSplat on large scenes.** Our method is able to reconstruct large-scale scenes, *e.g.*, scenes containing multiple rooms, as it is not constrained by the number of the training views, and the network architecture is based on sparse convolutions. Even though with more training frames, QuickSplat could cost more time to optimize, it is still considerable faster than other state-of-the-arts.

Initializer	Optimization & Densification	Abs err↓	Acc (2cm)↑	Acc (5cm)↑	Chamfer↓	Time
Ours	2DGS w/o densify	0.0692	0.4650	0.7211	0.1571	39s
Ours	2DGS w/ densify	**0.0668**	0.4796	0.7338	0.1486	39s
Ours	Ours	0.0732	**0.5263**	**0.7674**	**0.1461**	26s

Table 3. **Ablation on optimization and densification.** We compare Quicksplat's optimizer and densifier with original 2DGS optimiziation (w/ and w/o densificaation) under similar time frame.

Initializer	Optimization & Densification	Abs err↓	Acc (2cm)↑	Acc (5cm)↑	Chamfer↓	Time
SfM	SAGS w/o densify	0.1292	0.2781	0.5093	0.2879	429s
Ours	SAGS w/o densify	0.0692	0.4724	0.7297	0.1633	253s
Ours	SAGS w/ densify	**0.0669**	0.4825	0.7381	0.1625	276s
Ours	Ours	0.0732	**0.5263**	**0.7674**	**0.1461**	26s

Table 4. **Combined with SAGS [4].** We show that our initializer can be easily integrated into other methods, resulting in improved performance. In addition, our learned densification and optimization are faster and more accurate than SAGS under the same initialization. (Note that we modified SAGS to output 2D Gaussian splats for surface reconstruction.)