

A Additional ablations – Tables 1 and 2

Random initialization We analyze the effect of having sparse point cloud initialization versus random initialization in our method on 11 DSLR scenes from ScanNet++ [31]. for random initialization we do 5000 iterations in our warmup stage, as opposed to the usual 3500. We show that our method maintains the robustness to random initialization similar to 3DGS-MCMC [13], and despite a drop in number of planar Gaussians, it achieves comparable depth and image quality metrics to our method when initialized with SfM sparse point cloud.

Table 1: **Ablation on initialization** – Our method is robust to random initialization and achieves comparable performance to when initialized with SfM point cloud.

| Method | PSNR \uparrow | SSIM \uparrow | LPIPS \downarrow | RMSE \downarrow | MAE \downarrow | AbsRel \downarrow | #primitives (%planar) |
|----------------------|-----------------|-----------------|--------------------|-------------------|------------------|---------------------|-----------------------|
| 3DGS-MCMC (SfM) | 23.38 | 0.87 | 0.24 | 0.41 | 0.24 | 0.26 | 1.13M |
| Ours (SfM) | 23.42 | 0.87 | 0.24 | 0.20 | 0.13 | 0.12 | 1.13M (31%) |
| Ours (Random) | 23.30 | 0.86 | 0.25 | 0.21 | 0.14 | 0.13 | 1.13M (21%) |

Full metrics set for ablation on design choices We provide the full set of metrics for ablation on design choices (described in section 4.3) in the table 2

Table 2: **Ablation on design choices** – Loss components and optimization strategy are critical, with simultaneous plane-Gaussian optimization causing significant drops. 2D Gaussian snapping greatly improves depth accuracy compared to regularization alternatives. Similarly, Gaussian relocation is essential.

| | PSNR \uparrow | LPIPS \downarrow | SSIM \uparrow | RMSE \downarrow | MAE \downarrow | AbsRel \downarrow |
|-----------------------------|-----------------|--------------------|-----------------|-------------------|------------------|---------------------|
| Full model | 26.83 | 0.27 | 0.86 | 0.25 | 0.18 | 0.09 |
| Loss design: | | | | | | |
| w/o \mathcal{L}_{TV} | 23.24 | 0.34 | 0.82 | 0.34 | 0.24 | 0.13 |
| w/o \mathcal{L}_{mask} | 24.02 | 0.32 | 0.83 | 0.62 | 0.53 | 0.29 |
| Optimization design: | | | | | | |
| w/o plane optimization | 21.08 | 0.37 | 0.80 | 0.54 | 0.43 | 0.24 |
| simult. joint optimization | 19.52 | 0.38 | 0.79 | 0.40 | 0.32 | 0.18 |
| 2D Gaussian design: | | | | | | |
| w/o snapping | 25.53 | 0.31 | 0.84 | 0.38 | 0.31 | 0.17 |
| reg. w/o snapping | 21.69 | 0.35 | 0.81 | 0.36 | 0.28 | 0.15 |
| w/o relocation | 20.00 | 0.37 | 0.80 | 0.59 | 0.50 | 0.28 |

B Additional video and 3D mesh results

We provide video renderings of RGB and depth for our method compared to baselines in <https://3dgaussianflats.github.io>. We further provide video renderings of RGB and depth for our method compared to baselines in a supplemental video. Video results best capture the significant enhancement of our approach over baselines in depth estimation and accurately modeling scene geometry.

C Additional qualitative results

We provide more qualitative evidence for the performance of our method compared to 2DGS [7], 3DGS [6] and 3DGS-MCMC [13] baselines on the ScanNet++ [31] dataset in figure 8. The results show how baselines particularly struggle with reconstructing accurate geometry for the textureless areas while our method significantly improves upon these methods in depth estimation and keeps the visual quality of images.

Further, we provide more visualization for our estimated planes on ScanNet++ [31] dataset, showcasing the perfect alignment of our planes with the detected planar surfaces in figure 9.

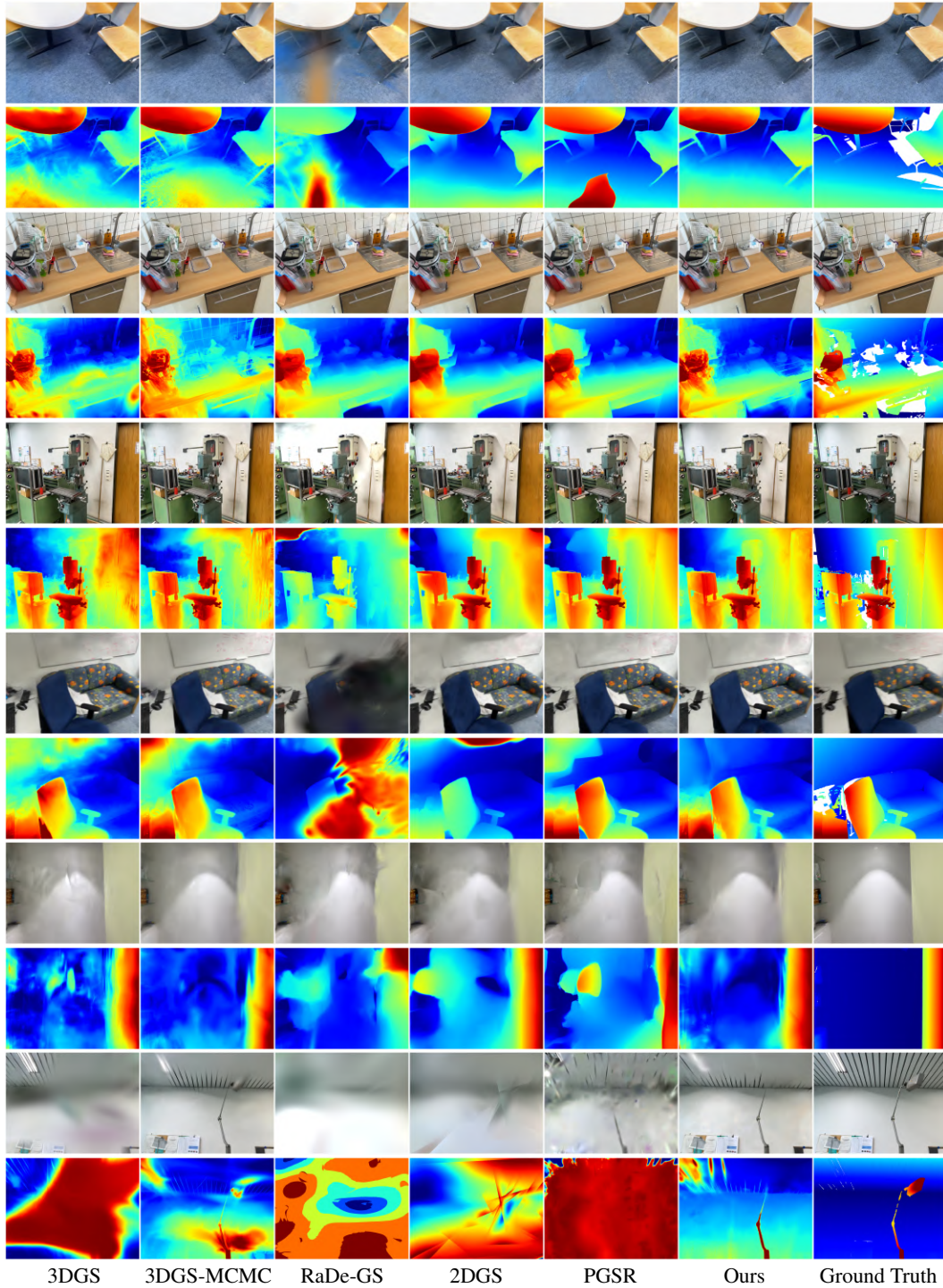


Figure 8: **Novel view synthesis and depth** – Qualitative results on ScanNet++ iPhone dataset show our superior performance in both image quality and depth estimation in novel views. Note the limitation of the quality of Gaussian Splatting based methods for textureless surfaces, which makes the plane fitting procedure less robust.

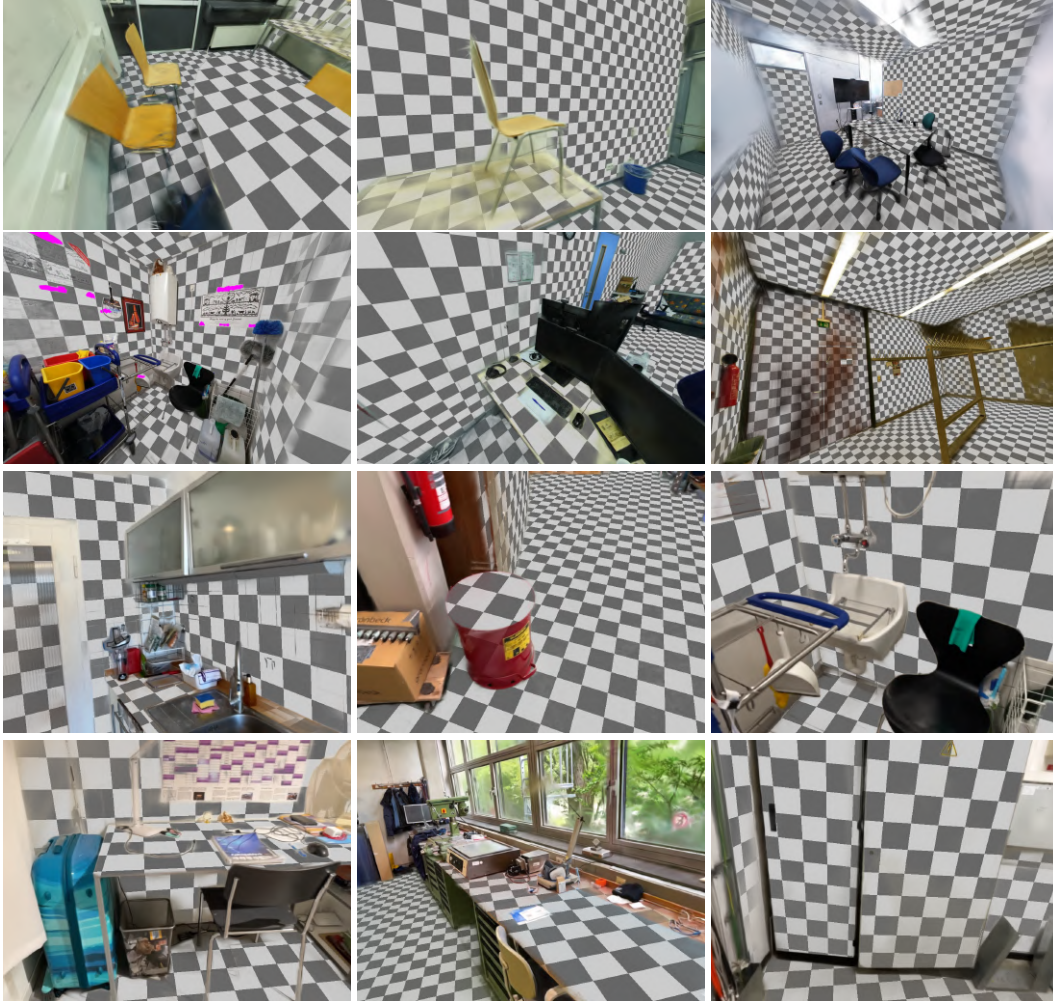


Figure 9: We provide visualizations of our output planes on the rendered test views of ScanNet++ DSLR streams (top 2 rows) and iPhone stream (bottom 2 rows). Pink markings are due to the anonymization of the original ScanNet++ dataset. While some planar surfaces are missed due to lack of manual 2D planar mask annotation, the captured planes are reconstructed faithfully.

D Input planar masks

2D semantic masks Our method relies on input consistent 2D segmentation masks of planar surfaces. To obtain these masks we can either annotate each image collection manually or automate the process for larger scenes. To automatically obtain the 2D segmentation masks, we employ PlaneRecNet [25] and SAMv2 video segmentation model [29], to create an annotation pipeline. We first input images to PlaneRecNet to obtain 2D plane annotations that are not semantically consistent across the image collection. We set the plane probability threshold to 0.5. While this method works well on iPhone images, it produces fewer plane annotations for DSLR images, that are out of distribution for its network trained on iPhone data. We then input these unmatched masks as seed to SAMv2. In order to do that, we order image collections that are not already sampled from a video. We propagate masks from the initial frame in 16-frame chunks of the sequence to the next 15 frames, and match SAMv2’s prediction with any subsequent 2D masks output from [25], using Hungarian matching with an IoU metric. Although largely effective, this process is prone to error accumulation through mask propagation. However, we assume resultant masks are semantically consistent across the image sequence. We provide sample segmentations of an input sequence in the supplementary video and on the website.

447 **Masked ground truth meshes** For the planar mesh extraction task, we only consider planes with
 448 annotated segmentation masks from the ground truth mesh, as the 2D plane segmentation task is
 449 orthogonal to our method. To identify the relevant subset of planes, we unproject points from the
 450 ground truth depth maps that correspond to each plane according to its segmentation mask. We then
 451 fit a plane to each resulting point cloud using RANSAC and compile these fitted planes into a set S .
 452 We match planes from the ground truth mesh to those in set S by applying two criteria: the normal
 453 cosine distance must be less than 0.99 to at least one plane in S , and the distance between their closest
 454 points must be less than 0.1. Doing this allows for computational efficiency and increased robustness
 455 to missing or undersegmented planes in the input 2D annotations.

456 **Code** We will release our code publicly for reproducibility purposes and to facilitate future research
 457 in this area. We base our code on the 3DGS-MCMC paper [13] and additionally use SAMv2 [29],
 458 and PlaneRecNet [25] to generate masks. The baselines are evaluated using their official released
 459 code [7, 6, 16, 17, 13, 8, 9]. We further utilize AirPlanes [9] code to compute meshing metrics.

460 E Hyperparameters settings

461 We use σ_{\perp} and σ_{\parallel} as hyper-parameters that control the stochastic re-location. These parameters are
 462 chosen depending on the metric scale of the dataset, and are defined in millimeters. For both datasets
 463 we used $\sigma_{\perp} = 0.01$ and $\sigma_{\parallel} = 0.3$. We observe that setting $\lambda_{\text{mask}} = 0.1$, yields best results empirically.
 464 For regularizers, we use $\lambda_{\text{TV}}=0.1$, $\lambda_{\text{scale}}=0.01$ and $\lambda_{\text{opacity}} = 0.01$ following [10] and [13]. We use
 465 the same scheduling policy for learning plane origin and normal (rotation) as for the Gaussian
 466 means the vanilla 3DGS. All experiments were conducted on a single A6000 ADA GPU, with 46GB
 467 memory. The method runs for approximately 1 hour for a single ScanNet++/ScanNetV2 scene,
 468 which is comparable to PGSR [16], the second best method for geometric quality according to our
 469 experiments and 1.5× longer than 3DGS-MCMC [13], the best method for Novel View Synthesis. The
 470 training time is increased due to the RANSAC overhead and block-coordinate descent optimization
 471 of planar parameters. Additionally, mesh extraction takes ~ 3 minutes and SAM mask propagation is
 472 on average 7 minutes long, depending on the scene type. We believe that the training time can be
 473 reduced in future work with addition of customized CUDA kernels.