
PlanarGS: High-Fidelity Indoor 3D Gaussian Splatting Guided by Vision-Language Planar Priors

Xirui Jin*

Shanghai Jiao Tong University

Renbiao Jin*

Shanghai Jiao Tong University

Boying Li[†]

Monash University
boying.li@monash.edu

Danping Zou[†]

Shanghai Jiao Tong University
dpzou@sjtu.edu.cn

Wenxian Yu

Shanghai Jiao Tong University

Abstract

Three-dimensional Gaussian Splatting (3DGS) has recently emerged as an efficient representation for novel-view synthesis, achieving impressive visual quality. However, in scenes dominated by large and low-texture regions, common in indoor environments, the photometric loss used to optimize 3DGS yields ambiguous geometry and fails to recover high-fidelity 3D surfaces. To overcome this limitation, we introduce PlanarGS, a 3DGS-based framework tailored for indoor scene reconstruction. Specifically, we design a pipeline for Language-Prompted Planar Priors (LP3) that employs a pretrained vision-language segmentation model and refines its region proposals via cross-view fusion and inspection with geometric priors. 3D Gaussians in our framework are optimized with two additional terms: a planar prior supervision term that enforces planar consistency, and a geometric prior supervision term that steers the Gaussians toward the depth and normal cues. We have conducted extensive experiments on standard indoor benchmarks. The results show that PlanarGS reconstructs accurate and detailed 3D surfaces, consistently outperforming state-of-the-art methods by a large margin. Project page: <https://planargs.github.io>

1 Introduction

Image-based 3D modeling is a long-standing challenge in the computer vision community, aiming to recover the 3D scene from a collection of images [42]. It serves as a fundamental component for a wide range of applications, including computer graphics, augmented/virtual reality (AR/VR), and robotics. Recently, 3D Gaussian Splatting (3DGS) [18] has emerged as a promising approach to modeling the 3D world. Given pictures captured from different viewpoints, 3DGS optimizes both the appearance and the geometry (scale, rotation, and position) of 3D Gaussian spheres to enforce the rendered image to be as close as possible to the real image captured in the target viewpoint. The appearance similarity, described by a photometric loss, plays a key role in the 3DGS training process.

Despite its impressive visual quality of view rendering, 3DGS still struggles to reconstruct a high-fidelity 3D surface model in scenes where large low-texture regions exist. Such kind of scenes are commonly seen in indoor environments. Lack of textures makes using the appearance cue as the supervision signal alone difficult to resolve the geometric ambiguities, resulting in inaccurate or false geometric reconstruction. To resolve such ambiguity, the recent method PGSR [4] applies the multi-view geometric consistency constraint to enhance the smoothness in areas with weak textures.

*Equal contribution.

[†]Corresponding author.

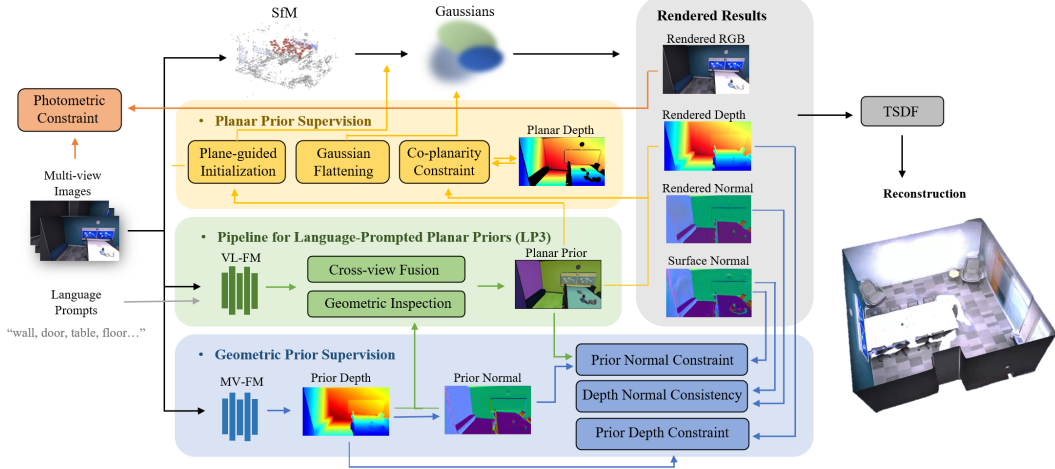


Figure 1: **PlanarGS overview.** Our method takes multi-view images and language prompts as inputs, getting planar priors through the pipeline for Language-Prompted Planar Priors (LP3). Our planar prior supervision includes plane-guided initialization, Gaussian flattening, and the co-planarity constraint, accompanied by geometric prior supervision. Both foundation models in the figure are pretrained.

Other approaches [6, 46, 12] have incorporated monocular depth or normal priors in the optimization process. However, existing methods still cannot handle large and texture-less planar regions. Those methods tend to produce locally smooth and globally curvy surfaces instead of perfect planes for those planar regions. One feasible solution is to explicitly detect those planar regions and enforce the planar geometry on these regions. This idea has been explored in monocular depth estimation [22, 61, 36] and Neural Radiance Field (NeRF) [13]. Those methods, however, highly rely on the performance of a heuristic plane detector [22] or a standalone neural plane detector [13] trained from limited data. However, such kinds of plane detectors often produce noisy segmentation and are difficult to apply to novel scenes (e.g, unseen planar objects not labeled in the training dataset).

In this paper, we introduce a novel framework of 3D Gaussian Splatting, **PlanarGS**, which leverages pretrained foundation models to overcome the aforementioned challenges of indoor 3D reconstruction. Foundation models, typically trained on vast, diverse datasets, exhibit far greater generalization and better performance than smaller, task-specific networks. To detect the planar regions in the image, we employ a pretrained vision-language segmentation model [39] prompted with terms such as "wall", "door", and "floor". Using word prompts enables flexible adaptation to new scenes by simply swapping or adding keywords, so the model can identify novel planar surfaces. For instance, in the classroom, one might include the extra word "blackboard" as the prompt to capture the novel planar structures not found in a home environment.

The detection results by the vision-language segmentation model always contain misleading region proposals, so we design the pipeline for Language-Prompt Planar Priors (**LP3**) to obtain reliable planar priors. For example, it is common that two connected perpendicular walls are merged into a single planar region. To resolve such ambiguities, additional geometric cues are necessary. Moreover, per-image detection frequently omits certain planes because it relies solely on local image information. Incorporating multi-view context can recover these missing labels by aggregating evidence from adjacent frames. Accordingly, we employ DUST3R [50], a pretrained multi-view 3D foundation model, to extract the depth and normal maps from neighboring images. Those maps serve as both the extra cues for more accurate planar region identification and as the geometric priors to guide the 3DGS optimization. We opt for DUST3R’s multi-view approach rather than a monocular alternative such as [56, 57] because it produces view-consistent 3D structures, enabling robust fusion of region proposals across views and yielding stronger 3D priors for the optimization process.

To incorporate planar priors into detected planar regions, we introduce a planar supervision scheme during 3DGS optimization. This includes plane-guided initialization, Gaussian flattening, and a co-planarity constraint, which together help preserve global surface flatness and suppress local artifacts.

Depth and normal priors from the pretrained 3D model are applied to under-constrained regions. We impose a prior normal constraint for planar regions, encouraging the surface normals extracted from 3D Gaussians to align with the per-pixel normals predicted by DUS3R. For low-texture regions, we constrain the rendered depth to match the scale-and-shift corrected prior depth and enforce consistency between the predicted depth and normal. Finally, to compensate for insufficient SfM points in texture-less planes, we densely initialize Gaussians by randomly sampling pixels within each detected plane and using their 3D coordinates, which are back-projected from the depth priors, as initial Gaussian centers.

Our main contributions include:

- We introduce a novel 3DGS framework that integrates semantic planar priors and multi-view depth priors to achieve high-fidelity reconstruction in challenging indoor scenes.
- We design a robust pipeline for Language-Prompted Planar Priors (LP3) that applies vision-language foundation models with cross-view fusion and geometric inspection to obtain reliable planar priors.
- Our method achieves state-of-the-art performance on high-fidelity surface reconstruction tasks across the Replica [44], ScanNet++ [60], and MuSHRoom [40] datasets.

2 Related Work

Image-based 3D Modeling: Image-based 3D modeling has been a long-standing goal in computer vision. The typical pipeline is as follows. Firstly, structure-from-motion (SfM) [42] is used to recover camera poses and sparse 3D points by matching feature points across different views. Then, based on the posed images, Multi-view stereo (MVS) [2, 43, 58, 32] generates dense depth maps from multi-view images via dense image matching, which are then converted to dense point clouds. Finally, surface reconstruction [34, 17] is applied to recover the 3D mesh and its texture from the point clouds, yielding the final 3D representation of the scene. This pipeline, successfully applied to outdoor photogrammetry for decades, still struggles to reconstruct accurate 3D models in texture-less scenes, such as indoor environments that usually contain large, texture-less planar regions. Moreover, using textured meshes for 3D scene representation often exhibits holes or other artifacts when rendered from novel viewpoints. Recently, implicit and point-based representations such as Neural Radiance Fields (NeRF) and Gaussian Splatting have emerged, offering superior expressiveness for jointly modeling geometry and appearance.

Neural Radiance Field (NeRF): Neural Radiance Field (NeRF) [33] utilizes a multilayer perceptron (MLP) to model a 3D scene as a continuous 5D radiance field optimized via volumetric rendering of posed images. However, because NeRF’s training also relies on photometric cues, it still struggles to recover accurate geometry in texture-less regions. To address this limitation, existing approaches have explored incorporating various priors. Many of them [48, 62, 45, 9, 35] use data-driven monocular depth or normal prediction as priors to improve the quality of the results. Other approaches incorporate indoor structural cues into reconstruction. For example, Manhattan-SDF [13] enforces a Manhattan-world assumption [11] by constraining floors and walls to axis-aligned planes, and another method [70] relaxes the constraints to the Atlantaworld assumption. Although better results have been achieved, their methods depend on a standalone plane detector trained on limited datasets, which reduces the flexibility across diverse scenes. Moreover, Manhattan-SDF requires an additional MLP to fuse noisy plane segmentations across views by comparing predicted labels from this MLP to observations. It is, however, not easy to extend such MLP fusion strategies to Gaussian Splatting, which represents scenes purely with spherical Gaussians.

Gaussian Splatting: Comparing with NeRF, 3D Gaussian Splatting [18] represents scenes explicitly with 3D Gaussians, which is more flexible and interpretable, as well as achieves significantly faster training and rendering speed. Similar to NeRF, 3DGS also depends on photometric loss for training. It therefore faces the same challenge in the texture-less environments. To enhance geometry reconstruction in Gaussian Splatting, numerous approaches [26, 46, 6, 12] also utilize monocular depth or normal priors for optimization. PGSR [4] introduces multi-view consistency, and FDS [7] leverages optical flow priors, both demonstrating performance gains. In addition, structure cues of indoor scenes have been exploited to guide Gaussian shapes and densification [1, 66, 41, 27]. Some methods jointly optimize Gaussian reconstruction with other representations, which are effective but

cumbersome. For example, GaussianRoom [53] combines with neural SDF [49], while other works incorporate textured mesh [16] or semantic maps [52, 38].

However, the aforementioned methods either focus on constraints of local regions or lack effective optimization mechanisms, making it difficult to reconstruct globally smooth surfaces. Our method designs a pipeline that adapts foundation models to obtain accurate planar priors, allowing for the holistic enforcement of co-planarity constraints in 3DGS. Meanwhile, multi-view depth priors also provide geometric scale information for PlanarGS.

3 Method

The whole pipeline of PlanarGS is illustrated in Fig. 1. Firstly, we apply a pretrained multi-view 3D foundation model [50] to obtain depth and normal priors. Subsequently, We leverage a pretrained vision-language foundation model [39] to the input images. Planar prior masks are generated from the model’s outputs through our LP3 pipeline (Sec. 3.1 **Pipeline for Language-Prompted Planar Priors**), which implements cross-view fusion and geometric verification using the depth and normal priors. In addition, the depth and normal priors serve to guide 3DGS optimization (Sec. 3.4 **Geometric Prior Supervision**). Notably, we propose a planar supervision mechanism (Sec. 3.3 **Planar Prior Supervision**) that flattens Gaussians and utilizes the planar priors to guide initialization and enforce the co-planarity constraint during optimization. Finally, the high-fidelity 3D mesh can be recovered from the rendered depth maps from PlanarGS via TSDF fusion [31, 34].

3.1 Preliminary: 3D Gaussian Splatting

Three-Dimensional Gaussian Splatting [18] represents scenes with a set of Gaussians $\{G_i\}$ initialized by the sparse SfM result. Each Gaussian is defined by the center μ_i and a full 3D covariance matrix Σ_i that can be factorized into a scaling matrix S_i and a rotation matrix R_i :

$$G_i(\mathbf{x}) = e^{-\frac{1}{2}(\mathbf{x}-\mu_i)^T \Sigma_i^{-1}(\mathbf{x}-\mu_i)} \quad \Sigma_i = R_i S_i S_i^T R_i^T. \quad (1)$$

To project 3D Gaussians to 2D for rendering, the center and the covariance matrix of Gaussian G_i can be projected to 2D coordinates as [71]:

$$\mu'_i = \mathbf{K} \mathbf{W} [\mu_i, 1]^T \quad \Sigma'_i = \mathbf{J} \mathbf{W} \Sigma_i \mathbf{W}^T \mathbf{J}^T, \quad (2)$$

where $\mathbf{W}$ is the viewing transformation matrix and $\mathbf{J}$ is the Jacobian of the affine approximation of the projective transformation. Rendering color C of a pixel can be given by α -blending [20]:

$$\hat{C} = \sum_{i \in N} T_i \alpha_i c_i \quad T_i = \prod_{j=i}^{i-1} (1 - \alpha_j), \quad (3)$$

where c_i and α_i represents color and density of the Gaussian and N is the number of Gaussians the ray passes through.

3.2 Pipeline for Language-Prompted Planar Priors

Existing plane segmentation methods [28, 55, 68, 65, 54, 51, 29] primarily design small and dedicated networks, which rely heavily on extensive annotated datasets and suffer from inaccurate segmentation. While they can be applied for coarse plane extraction and reconstruction, their results cannot serve as planar priors that impose strong constraints for high-quality 3D reconstruction, as ZeroPlane [29] in Fig. 5. By comparison, our pipeline for Language-Prompted Planar Priors (abbreviated as **LP3**) employs a vision-language foundation model GroundedSAM [39], applying cross-view fusion and geometric verification to its segmentation outputs to generate planar priors, as in Fig. 2. Simultaneously, since 3DGS is trained specifically for every scene, the tunability of prompts allows PlanarGS to achieve better reconstruction in atypical scenes.

Cross-view Fusion: We first provide text prompts to the object detection foundation model [30] and get bounding boxes. However, we found that large planes often go undetected in single images because their regions either span beyond the frame or lie close to the image boundaries. As shown in Fig. 2(a), to address the plane omission problem, we utilize the multi-view consistency from neighboring viewpoints for the cross-view fusion of plane boxes. Specifically, we leverage prior depth maps D_r (see Sec. 3.4) to back-project the main plane mask pixels p_s from a neighboring frame (source frame) into 3D points P_s :

$$P_s = D_r(p_s) \cdot \mathbf{K}^{-1} \tilde{p}_s, \quad (4)$$

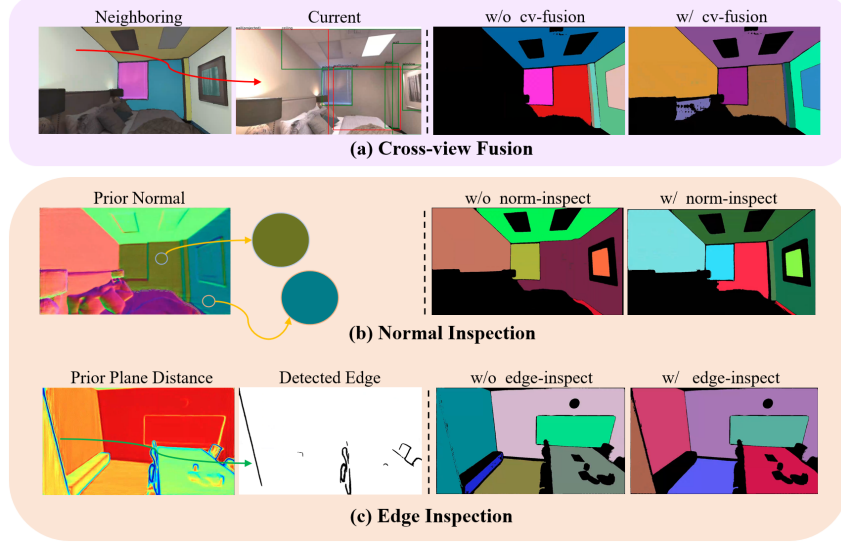


Figure 2: **Pipeline for Language-Prompted Planar Priors (LP3)**. (a) We employ cross-view fusion to supplement bounding box proposals. (b)(c) Prior normal and plane-distance maps are incorporated for geometric inspection. Consequently, we obtain abundant and accurate planar priors from this robust pipeline.

where $\mathbf{K}$ is the intrinsic matrix, $\mathbf{P}_s = [x, y, z]^T$ is in the camera coordinate system of source frame, $\tilde{\mathbf{p}}_s = [u, v, 1]^T$ is the homogeneous coordinate of the normalized pixel $\mathbf{p}_s$ in the source camera. $\mathbf{P}_s$ is then transformed into the camera coordinate system of the current frame (target frame) $\mathbf{P}_t$ and finally re-projected to 2D pixels $\mathbf{p}_t$:

$$\mathbf{P}_t = \mathbf{R}_t \mathbf{R}_s^T \mathbf{P}_s + (\mathbf{t}_t - \mathbf{R}_t \mathbf{R}_s^T \mathbf{t}_s) \quad z_t \tilde{\mathbf{p}}_t = \mathbf{K}_t \mathbf{P}_t. \quad (5)$$

After filtering out nested boxes with the same labels, we obtain refined detection results for downstream segmentation.

Geometric Inspection: Subsequently, we generate pixel-scale segmentation masks through the foundation model [19] according to the detected boxes. Nevertheless, sometimes the segmentation masks may incorrectly cover multiple planes together due to imprecise boxes, so we employ inspection from geometric priors as shown in Fig. 2(b)(c). We first convert depth priors D_r into surface-normal priors N_{dr} using the local plane assumption [37, 4]. For each pixel $\mathbf{p}$ in the depth map, we sample its four neighboring pixels and project these pixels into 3D points as Eq. 4, then the surface normal of the local plane at pixel $\mathbf{p}$ is:

$$N_{dr}(\mathbf{p}) = \frac{(\mathbf{P}_1 - \mathbf{P}_0) \times (\mathbf{P}_3 - \mathbf{P}_2)}{|(\mathbf{P}_1 - \mathbf{P}_0) \times (\mathbf{P}_3 - \mathbf{P}_2)|}. \quad (6)$$

We then express the distance from the local plane to the camera as $\delta_r(\mathbf{p}) = \mathbf{P} \cdot \mathbf{N}_{dr}(\mathbf{p})$ and get a prior plane-distance map δ_r . We apply the K-means clustering to partition the prior normal map N_{dr} to separate non-parallel planes. For the separation of parallel planes, we identify the outliers of the plane-distance map as geometric edges that violate the local plane assumption. This stage also splits non-planar areas into fragments so we can filter them out.

3.3 Planar Prior Supervision

Inspired by previous works [61, 22, 21, 23], we leverage semantic planar priors as a holistic supervision for indoor reconstruction. Unlike existing approaches [13, 70, 8, 59, 64] that incorporate semantic features, our method treats the planar prior generation as a pre-processing stage (see Sec. 3.2), offering a simpler solution.

Plane-guided Initialization: The Gaussian initialization based on SfM relies on feature extraction, leading to point cloud deficiency in low-texture regions. We back-project pixels of plane areas to dense 3D points $\mathbf{P}_m$ as Eq. 4, while obtaining the plane labels of existing Gaussian points by

projecting them to 2D. For each plane region in each viewpoint, we calculate the average distance of existing Gaussians, deciding whether to supplement it with dense 3D plane points P_m .

Gaussian Flattening and Depth Rendering: In geometric reconstruction, flattening the Gaussians into planes enables them to conform more closely to the real-world surfaces, meanwhile producing less ambiguous depth and normal renderings [4]. Accordingly, we minimize the minimum scale factor $S_i = \text{diag}(s_1, s_2, s_3)$ for each Gaussian [5] and define the minimum scale factor as the normal of the Gaussian n_i . The normal map of the current viewpoint can be rendered through α -blending with the rotation matrix from the camera to the global world R_c :

$$L_s = \|\min(s_1, s_2, s_3)\|_1 \quad \hat{N} = \sum_{i \in N} R_c^T n_i \alpha_i \prod_{j=i}^{i-1} (1 - \alpha_j). \quad (7)$$

For each flattened Gaussian, the plane distance d_i to the camera center O_c can be expressed by projecting the distance of the Gaussian center μ_i to the normal direction n_i . The rendered plane-distance map $\hat{\delta}$ is also obtained through α -blending:

$$d_i = (R_c^T (\mu_i - O_c)) R_c^T n_i^T \quad \hat{\delta} = \sum_{i \in N} d_i \alpha_i \prod_{j=i}^{i-1} (1 - \alpha_j). \quad (8)$$

Following method [4], we obtain the rendered depth map $\hat{D}$ of Gaussians from the rendered plane-distance $\hat{\delta}$ and normal map $\hat{N}$:

$$\hat{D}(p) = \frac{\hat{\delta}(p)}{\hat{N}(p) K^{-1} \tilde{p}}, \quad (9)$$

where $p = [u, v]^T$ is the pixel position and $\tilde{p}$ is the homogeneous coordinate of p .

Co-planarity Constraint: To incorporate the holistic planar priors into Gaussian optimization, we apply the co-planarity constraint on the depth map. We first back-project the rendered depth map $\hat{D}$ to dense 3D points as Eq. 4. Following methods [61, 22], we define the plane p_m in the 3D space as $A_m^T P = 1$, where A_m is the plane parameter and its direction represents the normal of the plane. As a result, the dense 3D points within the plane region p_m should follow an equation $Q_n A_m = Y_m$ and A_m can be obtained by solving the least squares problem:

$$A_m = (Q_n^T Q_n + \epsilon E)^{-1} Q_n^T Y_m, \quad (10)$$

where $Q_n = [P_1, P_2 \dots P_n]^T$, $Y_n = [1, 1 \dots 1]^T$ and ϵ is a small scale to ensure numerical stability with an identity matrix E . Subsequently, we obtain the planar depth map D_p as:

$$D_p(p) = (A_m^T K^{-1} \tilde{p})^{-1}. \quad (11)$$

The depth D_p obtained from plane fitting is then used as a co-planarity constraint to the rendered depth. The loss function is defined as:

$$L_p = \frac{1}{N_p} \sum_{p \in p} \|D_p(p) - \hat{D}(p)\|_1, \quad (12)$$

where p represents all plane regions and N_p represents the number of pixels in plane regions. Curved objects and clutter in the scenes are effectively excluded by our planar masks. Since these objects usually exhibit rich textures and geometric details, their reconstruction is less problematic.

3.4 Geometric Prior Supervision

Previous methods mostly introduce depth priors from monocular depth estimation, which suffer from local misalignments [67, 14, 24, 25]. We instead utilize the pretrained multi-view foundation model [50] and provide view-consistent geometric supervision of Gaussian optimization together with the planar priors.

Prior Depth Constraint: We align the resized dense depth map D_{dense} obtained from DUST3R with the sparse depth D_{sparse} derived by projecting the SfM-reconstructed point cloud to the current viewpoint. Because of the memory consumption, the depth maps are generated in groups, and each group shares the same scale and shift parameters.

$$s^*, t^* = \arg \min_{s, t} \sum_{p \in D_{sparse}} \|D_{sparse}(p) - (s \cdot D_{dense}(p) + t)\|_1, \quad (13)$$

and we utilize the aligned dense depth map as depth prior: $D_r = s^* \cdot D_{dense} + t^*$. We leverage the prior depth as a constraint on the rendered depth:

$$L_{rd} = \frac{1}{N_{lt}} \sum_{p \in lt} M_{cof}(p) \cdot \|D_r(p) - \hat{D}(p)\|^2, \quad (14)$$

where M_{cof} is the mask from the confidence map of DUST3R for its depth prediction. The lt represents the collection of pixels in low-texture regions, generated by computing a mask through the Canny edge detector [3] on RGB input to mitigate the effect of depth priors' low resolution.

Prior Normal Constraint: The prior depth constraint serves as a scale-based supervision, while normal supervision offers greater flexibility in certain scenes and has been widely adopted in recent studies [4, 6, 12]. Following Eq. 4 and Eq. 6, we can obtain the surface-normal map $\hat{N}_d$ from the rendered depth $\hat{D}$. We constrain the rendered surface-normal $\hat{N}_d$ with the prior surface-normal N_{dr} :

$$L_{rn} = \|N_{dr} - \hat{N}_d\|_1 + (1 - N_{dr} \cdot \hat{N}_d) \quad (p \in p). \quad (15)$$

The normal prior is exclusively utilized on the plane region p we extracted, optimizing the position of Gaussians together with the co-planarity constraint in Sec. 3.3.

Depth Normal Consistency: Additionally, inspired by works [4, 6, 12], we introduce the depth normal consistent regularization between rendered GS-normal $\hat{N}$ and rendered surface-normal $\hat{N}_d$ to optimize the rotation of Gaussians consistent with their positions:

$$L_{dn} = \frac{1}{N_{lt}} \sum_{p \in lt} \cdot \|\hat{N}_d(p) - \hat{N}(p)\|_1, \quad (16)$$

where lt is the same as Eq. 14 to avoid edge areas that violate the local plane assumption in Eq. 6.

3.5 Optimization

The overall loss function of PlanarGS that combines planar and geometric supervision is:

$$L_{total} = L_{RGB} + L_s + \lambda_1 L_{dn} + \lambda_2 L_p + \lambda_3 L_{rd} + \lambda_4 L_{rn}, \quad (17)$$

where λ are parameters and L_{RGB} includes L_1 and D-SSIM loss as 3DGS.

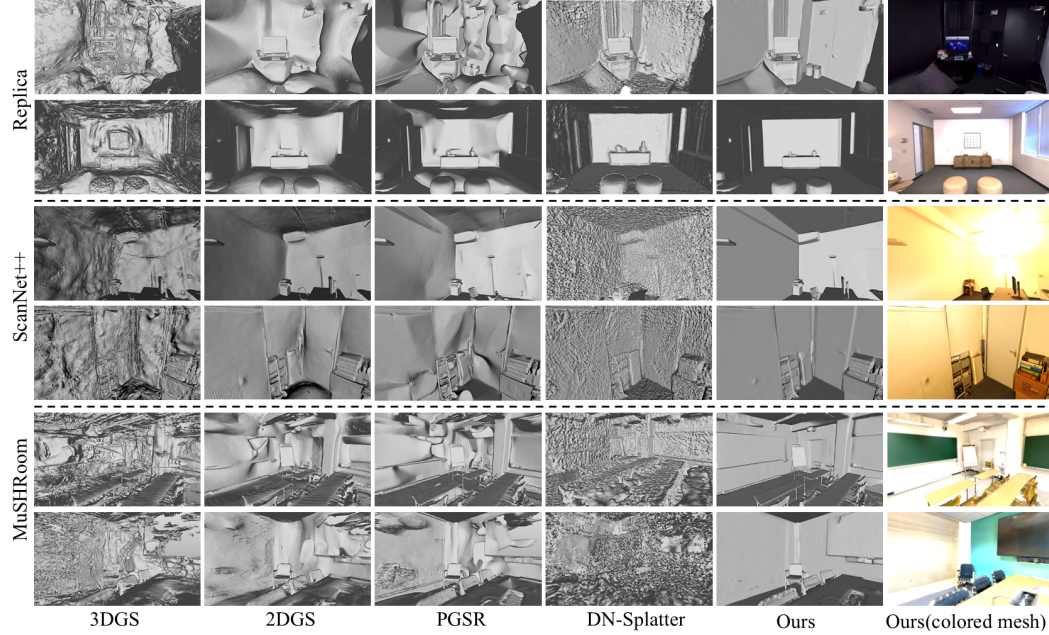


Figure 3: **Qualitative comparison.** We present reconstructed meshes from other methods and PlanarGS. The right column shows our colored meshes. The results demonstrate that PlanarGS achieves more accurate and comprehensive high-fidelity mesh reconstruction.

4 Experiments

We demonstrate the superior performance of PlanarGS on 3D reconstruction of indoor scenes in Sec. 4.1 and conduct ablation studies in Sec. 4.2.

Datasets: Following other works [41, 46] on Gaussian reconstruction in indoor scenes, we conduct our experimental evaluation on three indoor scene datasets, including 8 scenes from the synthetic dataset Replica [44] and 9 scenes from the real-world datasets: 4 scenes from ScanNet++ [60] and 5 complex scenes from MuSHRoom [40]. We use COLMAP [42] for all these scenes to generate a sparse point cloud as initialization.

Implementation Details: All experiments in this paper are conducted on the Nvidia RTX 3090 GPU. The training iterations for all scenes are set to 30,000, and $\lambda_1, \lambda_2, \lambda_3, \lambda_4$ are set as 0.05, 0.5, 0.05, 0.2, respectively. We use "wall, floor, door, screen, window, ceiling, table" as prompts for planar prior generation on Replica and ScanNet++ datasets, as these classes are commonly found in various indoor environments, and additionally add "blackboard, sofa" on the MuSHRoom dataset.

Evaluation Metrics: Consistent with existing methods [46, 7, 38] to evaluate the quality of reconstructed surfaces, we report Accuracy (Acc), Completion (Comp), and Chamfer Distance (CD) in cm. F-score (F1) with a 5cm threshold and Normal Consistency (NC) are reported as percentages. For novel view synthesis (NVS), we follow standard PSNR, SSIM, and LPIPS metrics for rendered images.

4.1 Reconstruction Results

We conduct comparisons with the baseline 3DGS [18], and Gaussian surface reconstruction methods 2DGS [15], GOF [63], PGSR [4], QGS [69], DN-Splatter [46]. For 3DGS, we obtain depth maps by applying alpha-blending integration to the z-coordinates of the Gaussians and generate a mesh through TSDF. For DN-Splatter, we apply depth priors from the multi-view foundation model that same as ours.

Quantitative Results: The quantitative comparison on three datasets are presented in Tab. 1 and Tab. 2. The reconstruction results of our method significantly outperform those of other methods across benchmarks, while simultaneously ensuring high-quality novel view synthesis in Tab. 1. Additionally, PlanarGS requires training time within one hour, comparable to other 3DGS-based methods. We also generate meshes directly from the depth priors provided by DUST3R using TSDF, as shown in Tab. 2, to demonstrate the high-fidelity advantage of Gaussian splatting methods.

Table 1: Quantitative results of surface reconstruction and NVS on **MuSHRoom** dataset. Our method outperforms other methods on most metrics. The best results are marked in **bold**.

	Surface Reconstruction					NVS		
	Acc↓	Comp↓	CD↓	F1↑	NC↑	PSNR↑	SSIM↑	LPIPS↓
3DGS [18]	12.01	11.85	11.92	38.53	62.00	25.79	0.8775	0.2059
2DGS [15]	9.16	10.27	9.71	51.50	73.65	25.67	0.8744	0.2265
GOF [63]	15.65	8.50	12.07	42.57	61.47	24.76	0.8646	0.2189
PGSR [4]	7.52	13.50	10.51	59.11	73.27	25.73	0.8761	0.2250
QGS [69]	8.53	10.02	9.28	55.15	73.66	24.37	0.8624	0.1994
DN-Splatter [46]	6.25	5.29	5.77	61.86	77.13	24.80	0.8549	0.2595
Ours	3.95	5.02	4.49	77.14	83.35	26.42	0.8874	0.2141

Table 2: Quantitative results of surface reconstruction on **ScanNet++** and **Replica** datasets. Our method outperforms other methods on most metrics. The best results are marked in **bold**.

	ScanNet++					Replica				
	Acc↓	Comp↓	CD↓	F1↑	NC↑	Acc↓	Comp↓	CD↓	F1↑	NC↑
DUST3R [50]	9.70	6.64	8.17	38.17	75.27	7.52	7.18	7.35	44.89	68.19
3DGS [18]	11.02	10.12	10.57	31.78	77.65	12.40	12.75	12.57	41.64	69.83
2DGS [15]	6.89	6.52	6.7	53.46	88.20	9.02	12.05	10.54	52.91	82.28
GOF [63]	7.76	6.42	7.09	57.99	74.75	9.77	8.31	9.04	54.08	74.34
PGSR [4]	7.32	7.13	7.22	53.73	87.30	7.32	9.79	8.56	62.98	83.30
DN-Splatter [46]	4.88	3.44	4.16	75.86	82.06	4.95	6.25	5.60	68.12	80.80
Ours	3.86	3.46	3.66	82.78	90.52	2.80	5.45	4.13	81.90	89.88

Qualitative Results: Fig. 3 shows mesh comparisons. While 3DGS [18] produces rough surfaces, 2DGS/PGSR’s [15, 4] flattened Gaussians create local smoothness but suffer from bending artifacts due to the missing of geometric constraints. Despite using depth and normal priors, DN-Splatter [46] still has inaccurate wall positioning in some scenes, and the lack of normal consistency causes roughness. Our method uniquely obtains both geometrically accurate planes and smooth surfaces with our planar and geometric priors, serving as a high-fidelity indoor reconstruction method. The quality of reconstruction is also demonstrated by the novel view rendering results shown in Fig. 4.



Figure 4: **Novel view synthesis comparison.** With the introduction of planar and geometric supervision, PlanarGS can effectively eliminate artifacts present in other methods.

4.2 Ablation Study

To validate the effectiveness of PlanarGS’s components and the robustness of the pre-processing stage, we conducted ablation studies on the MuSHRoom and Replica datasets.

Module Effectiveness: We begin with an ablation on planar prior generation, comparing against the task-specific SOTA model ZeroPlane [29] and the vision-language foundation model GroundedSAM [39] without cross-view fusion and geometric inspection. As in Tab. 3, in cases of inaccurate planar priors, co-planarity constraints may act on erroneous regions, leading to a reduction in reconstruction precision. A visual comparison of planar priors can be found in Fig. 5, where our planar priors effectively separate different planes and exclude non-planar clutter.

We further analyze the effect of each constraint in Gaussian optimization. When without the co-planarity constraint, as in Fig. 6, depth and normal priors provide geometric constraints only at the local scale, leading to surface irregularities in large plane regions. As can be seen from the Tab. 3, in the absence of geometric priors from pretrained models, the co-planarity constraint contributes more significantly to the reconstruction results. However, without geometric prior constraint, scale-aware supervision is missing, causing large planes under the co-planarity constraint to tilt and deviate as a whole, as in Fig. 6. Finally, when the consistency constraint between rendered normals and surface normals is absent, the flattened Gaussians fail to align with surfaces. This absence affects the normal accuracy of the mesh, displaying as rough surfaces particularly in regions without our co-planarity constraint, as shown in Tab. 3, Fig. 6.

Table 3: Ablation of module effectiveness on the coffee room of **MuSHRoom** dataset. The last row denotes our full model with all components enabled.

Pre-proc. Planar Prior	Opt. Constraints			Metrics				
	Co-planar.	Geom. Prior	DN Consist.	Acc↓	Comp↓	CD↓	F1↑	NC↑
ZeroPlane [29]	✓	✓	✓	4.76	5.73	5.25	62.49	82.41
GroundedSAM [39]	✓	✓	✓	2.95	3.92	3.43	80.22	84.66
LP3 pipeline	-	✓	✓	2.83	3.53	3.18	85.62	84.63
LP3 pipeline	✓	-	✓	4.63	5.01	4.82	73.96	80.62
LP3 pipeline	-	-	✓	6.93	6.26	6.59	64.11	74.42
LP3 pipeline	✓	✓	-	2.57	3.17	2.87	88.19	81.95
LP3 pipeline	✓	✓	✓	2.55	3.40	2.98	87.32	85.28

Model Robustness: To validate the robustness of our model, we perform extra ablations, as in Tab. 4. The multi-view foundation model can be substituted with VGGT [47], allowing this stage to be completed within a few seconds. We also replace the vision-language foundation model with YoloWorld [10] and SAM [19], and add "blackboard, sofa" in addition to regular prompts

for the Replica dataset. Both changes have minimal impact on the reconstruction results. These results highlight the robustness of our pipeline for Language-Prompted Planar Priors (LP3) and prior supervision, while indicating that improvements in foundation models may further boost the performance of our approach.

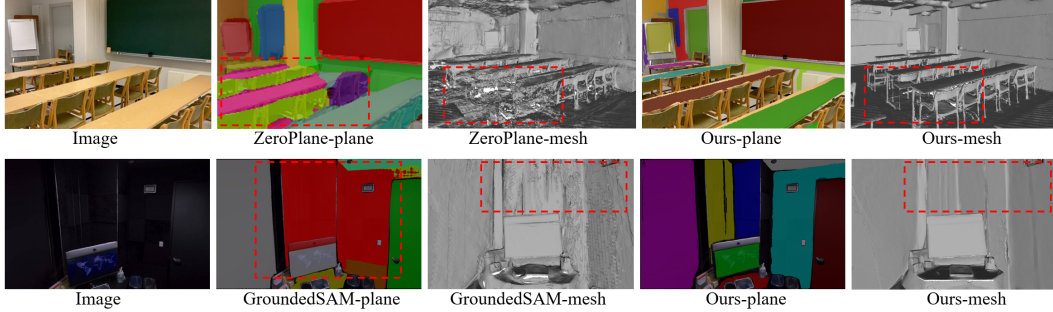


Figure 5: **Ablation of planar priors.** ZeroPlane tends to wrongly segment cluttered objects into planar regions, while using GroundedSAM without the pipeline for Language-Prompted Planar Priors (LP3) cannot distinguish different planes within a single object. Neither of them can provide reliable planar priors for Gaussian optimizing.

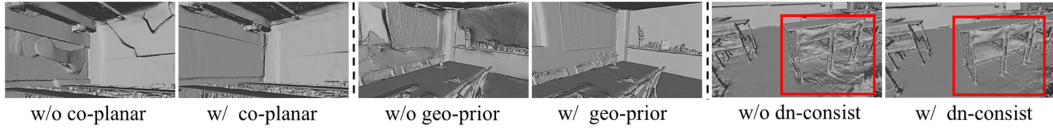


Figure 6: **Visualization of constraint effectiveness.** Left to right: results without and with co-planarity, geometric prior, and depth normal consistency constraints. Improvements are highlighted in red boxes. The enhancements show our proposed constraints lead to high-fidelity reconstruction.

Table 4: Ablation of model robustness on the **Replica** dataset. The last row shows the settings used for our model in the experiments.

MV-FM	Pre-processing VL-FM	Prompts	Metrics				
			Acc↓	Comp↓	CD↓	F1↑	NC↑
VGGT [47]	GroundedSAM	regular	3.96	6.32	5.14	75.41	88.85
DUST3R	YoloWorld [10]+SAM	regular	2.92	5.56	4.24	80.96	89.04
DUST3R	GroundedSAM	more prompts	2.88	5.50	4.19	81.43	89.32
DUST3R	GroundedSAM	regular	2.80	5.45	4.13	81.90	89.88

5 Conclusion

We present a novel method, PlanarGS, that incorporates planar and multi-view depth priors into 3DGS, addressing the inaccurate reconstruction of large and texture-less planes commonly seen in indoor scenes. Extensive experiments on Replica, ScanNet++, and MuSHRoom datasets demonstrate that our method outperforms existing approaches in reconstruction. The language-prompted planar priors make our method flexible to multiple scenes in various domains like VR and robotics.

Limitations: Our planar priors are only effective for detected plane regions, which work well in indoor environments dominated by large flat surfaces. However, these priors offer no improvement for the reconstruction of curved walls or natural outdoor scenes lacking man-made planar structures.

Acknowledgments and Disclosure of Funding

This work was supported by the National Key R&D Program of China (2022YFB3903801) and the National Science Foundation of China (62073214).

References

- [1] Jiayang Bai, Letian Huang, Jie Guo, Wen Gong, Yuanqi Li, and Yanwen Guo. 360-gs: Layout-guided panoramic gaussian splatting for indoor roaming. In *International Conference on 3d Vision (3DV)*, 2025.
- [2] Michael Bleyer, Christoph Rhemann, and Carsten Rother. Patchmatch stereo-stereo matching with slanted support windows. In *Proceedings of the British Machine Vision Conference*, volume 11, pages 1–11, 2011.
- [3] John Canny. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, pages 679–698, 1986.
- [4] Danpeng Chen, Hai Li, Weicai Ye, Yifan Wang, Weijian Xie, Shangjin Zhai, Nan Wang, Haomin Liu, Hujun Bao, and Guofeng Zhang. Pgsr: Planar-based gaussian splatting for efficient and high-fidelity surface reconstruction. *IEEE Transactions on Visualization and Computer Graphics*, 2024.
- [5] Hanlin Chen, Chen Li, and Gim Hee Lee. Neusg: Neural implicit surface reconstruction with 3d gaussian splatting guidance. *arXiv preprint arXiv:2312.00846*, 2023.
- [6] Hanlin Chen, Fangyin Wei, Chen Li, Tianxin Huang, Yunsong Wang, and Gim Hee Lee. Vcr-gaus: View consistent depth-normal regularizer for gaussian surface reconstruction. *Advances in Neural Information Processing Systems (NeurIPS)*, 37:139725–139750, 2024.
- [7] Lin-Zhuo Chen, Kangjie Liu, Youtian Lin, Zhihao Li, Siyu Zhu, Xun Cao, and Yao Yao. Flow distillation sampling: Regularizing 3d gaussians with pre-trained matching priors. In *International Conference on Learning Representations (ICLR)*, 2025.
- [8] Zheng Chen, Qingan Yan, Huangying Zhan, Changjiang Cai, Xiangyu Xu, Yuzhong Huang, Weihaan Wang, Ziyue Feng, Yi Xu, and Lantao Liu. Planarnerf: Online learning of planar primitives with neural radiance fields. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 13942–13949. IEEE, 2025.
- [9] Ziyi Chen, Xiaolong Wu, and Yu Zhang. Nc-sdf: Enhancing indoor scene reconstruction using neural sdfs with view-dependent normal compensation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5155–5165, 2024.
- [10] Tianheng Cheng, Lin Song, Yixiao Ge, Wenyu Liu, Xinggang Wang, and Ying Shan. Yolo-world: Real-time open-vocabulary object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16901–16911, 2024.
- [11] James M Coughlan and Alan L Yuille. Manhattan world: Compass direction from a single image by bayesian inference. In *Proceedings of the IEEE international conference on computer vision (ICCV)*, volume 2, pages 941–947. IEEE, 1999.
- [12] Pinxuan Dai, Jiamin Xu, Wenxiang Xie, Xinguo Liu, Huamin Wang, and Weiwei Xu. High-quality surface reconstruction using gaussian surfels. In *ACM SIGGRAPH 2024 Conference Papers*, pages 1–11, 2024.
- [13] Haoyu Guo, Sida Peng, Haotong Lin, Qianqian Wang, Guofeng Zhang, Hujun Bao, and Xiaowei Zhou. Neural 3d scene reconstruction with the manhattan-world assumption. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5511–5520, 2022.
- [14] Seungoh Han, Jaehoon Jang, Hyunsu Kim, Jaeheung Surh, Junhyung Kwak, Hyowon Ha, and Kyungdon Joo. Lighthouseg: Indoor structure-aware 3d gaussian splatting for panorama-style mobile captures. *arXiv preprint arXiv:2507.06109*, 2025.
- [15] Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2d gaussian splatting for geometrically accurate radiance fields. In *ACM SIGGRAPH 2024 conference papers*, pages 1–11, 2024.
- [16] Binxiao Huang, Zhihao Li, Shiyong Liu, Xiao Tang, Jiajun Tang, Jiaqi Lin, Yuxin Cheng, Zhenyu Chen, Xiaofei Wu, and Ngai Wong. Hybrid mesh-gaussian representation for efficient indoor scene reconstruction. *arXiv preprint arXiv:2506.06988*, 2025.
- [17] Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. Poisson surface reconstruction. In *Proceedings of the fourth Eurographics symposium on Geometry processing*, volume 7, 2006.
- [18] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics (TOG)*, 42(4):139–1, 2023.
- [19] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4015–4026, 2023.

- [20] Georgios Kopanas, Thomas Leimkühler, Gilles Rainer, Clément Jambon, and George Drettakis. Neural point catacaustics for novel-view synthesis of reflections. *ACM Transactions on Graphics (TOG)*, 41(6): 1–15, 2022.
- [21] Boying Li, Danping Zou, Daniele Sartori, Ling Pei, and Wenxian Yu. Textslam: Visual slam with planar text features. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2102–2108. IEEE, 2020.
- [22] Boying Li, Yuan Huang, Zeyu Liu, Danping Zou, and Wenxian Yu. Structdepth: Leveraging the structural regularities for self-supervised indoor depth estimation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 12663–12673, 2021.
- [23] Boying Li, Danping Zou, Yuan Huang, Xinghan Niu, Ling Pei, and Wenxian Yu. Textslam: Visual slam with semantic planar text features. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(1):593–610, 2023.
- [24] Boying Li, Zhixi Cai, Yuan-Fang Li, Ian Reid, and Hamid Rezatofighi. Hier-slam: Scaling-up semantics in slam with a hierarchically categorical gaussian splatting. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9748–9754. IEEE, 2025.
- [25] Boying Li, Vuong Chi Hao, Peter J Stuckey, Ian Reid, and Hamid Rezatofighi. Hier-slam++: Neuro-symbolic semantic slam with a hierarchically categorical gaussian splatting. *arXiv preprint arXiv:2502.14931*, 2025.
- [26] Jiahe Li, Jiawei Zhang, Xiao Bai, Jin Zheng, Xin Ning, Jun Zhou, and Lin Gu. Dngaussian: Optimizing sparse-view 3d gaussian radiance fields with global-local depth normalization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 20775–20785, 2024.
- [27] Yanyan Li, Chenyu Lyu, Yan Di, Guangyao Zhai, Gim Hee Lee, and Federico Tombari. Geogaussian: Geometry-aware gaussian splatting for scene rendering. In *Proceedings of the European conference on computer vision (ECCV)*, 2024.
- [28] Chen Liu, Kihwan Kim, Jinwei Gu, Yasutaka Furukawa, and Jan Kautz. Planercnn: 3d plane detection and reconstruction from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4450–4459, 2019.
- [29] Jiachen Liu, Rui Yu, Sili Chen, Sharon X Huang, and Hengkai Guo. Towards in-the-wild 3d plane reconstruction from a single image. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 27027–27037, 2025.
- [30] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *Proceedings of the European conference on computer vision (ECCV)*, pages 38–55. Springer, 2024.
- [31] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *Seminal graphics: pioneering efforts that shaped the field*, pages 347–353, 1998.
- [32] Wenjie Luo, Alexander G Schwing, and Raquel Urtasun. Efficient deep learning for stereo matching. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5695–5703, 2016.
- [33] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021.
- [34] Richard A Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J Davison, Pushmeet Kohi, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *2011 10th IEEE international symposium on mixed and augmented reality*, pages 127–136. Ieee, 2011.
- [35] Minyoung Park, Mirae Do, YeonJae Shin, Jaeseok Yoo, Jongkwang Hong, Joongrock Kim, and Chul Lee. H2o-sdf: two-phase learning for 3d indoor reconstruction using object surface fields. In *International Conference on Learning Representations (ICLR)*, 2024.
- [36] Vaishakh Patil, Christos Sakaridis, Alexander Liniger, and Luc Van Gool. P3depth: Monocular depth estimation with a piecewise planarity prior. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1610–1621, June 2022.

- [37] Xiaojuan Qi, Renjie Liao, Zhengzhe Liu, Raquel Urtasun, and Jiaya Jia. Geonet: Geometric neural network for joint depth and surface normal estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 283–291, 2018.
- [38] Jiaxiong Qiu, Liu Liu, Zhizhong Su, and Tianwei Lin. Gls: Geometry-aware 3d language gaussian splatting. *arXiv preprint arXiv:2411.18066*, 2024.
- [39] Tianhe Ren, Shilong Liu, Ailing Zeng, Jing Lin, Kunchang Li, He Cao, Jiayu Chen, Xinyu Huang, Yukang Chen, Feng Yan, Zhaoyang Zeng, Hao Zhang, Feng Li, Jie Yang, Hongyang Li, Qing Jiang, and Lei Zhang. Grounded sam: Asembling open-world models for diverse visual tasks. *arXiv preprint arXiv:2401.14159*, 2024.
- [40] Xuqian Ren, Wenjia Wang, Dingding Cai, Tuuli Tuominen, Juho Kannala, and Esa Rahtu. Mushroom: Multi-sensor hybrid room dataset for joint 3d reconstruction and novel view synthesis. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, page 4508–4517, 2024.
- [41] Cong Ruan, Yuesong Wang, Tao Guan, Bin Zhang, and Lili Ju. Indoorgs: Geometric cues guided gaussian splatting for indoor scene reconstruction. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 844–853, 2025.
- [42] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4104–4113, 2016.
- [43] Johannes L Schönberger, Enliang Zheng, Jan-Michael Frahm, and Marc Pollefeys. Pixelwise view selection for unstructured multi-view stereo. In *Proceedings of the European conference on computer vision (ECCV)*, pages 501–518. Springer, 2016.
- [44] Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, et al. The replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019.
- [45] Ziyu Tang, Weicai Ye, Yifan Wang, Di Huang, Hujun Bao, Tong He, and Guofeng Zhang. Nd-sdf: Learning normal deflection fields for high-fidelity indoor reconstruction. In *International Conference on Learning Representations (ICLR)*, 2025.
- [46] Matias Turkulainen, Xuqian Ren, Iaroslav Melekhov, Otto Seiskari, Esa Rahtu, and Juho Kannala. Dn-splatter: Depth and normal priors for gaussian splatting and meshing. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 2421–2431. IEEE, 2025.
- [47] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.
- [48] Jiepeng Wang, Peng Wang, Xiaoxiao Long, Christian Theobalt, Taku Komura, Lingjie Liu, and Wenping Wang. Neuris: Neural reconstruction of indoor scenes using normal priors. In *Proceedings of the European conference on computer vision (ECCV)*, pages 139–155. Springer, 2022.
- [49] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.
- [50] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20697–20709, 2024.
- [51] Jamie Watson, Filippo Aleotti, Mohamed Sayed, Zawar Qureshi, Oisin Mac Aodha, Gabriel Brostow, Michael Firman, and Sara Vicente. Airplanes: Accurate plane estimation via 3d-consistent embeddings. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5270–5280, 2024.
- [52] Xiaobao Wei, Xiaohan Zhang, Hao Wang, Qingpo Wu, Ming Lu, Wenzhao Zheng, and Shanghang Zhang. Omniindoor3d: Comprehensive indoor 3d reconstruction. *arXiv preprint arXiv:2505.20610*, 2025.
- [53] Haodong Xiang, Xinghui Li, Kai Cheng, Xiansong Lai, Wanting Zhang, Zhichao Liao, Long Zeng, and Xueping Liu. Gaussianroom: Improving 3d gaussian splatting with sdf guidance and monocular cues for indoor scene reconstruction. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024.

- [54] Yaxu Xie, Jason Rambach, Fangwen Shu, and Didier Stricker. Planesegnet: Fast and robust plane estimation using a single-stage instance segmentation cnn. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 13574–13580. IEEE, 2021.
- [55] Yaxu Xie, Fangwen Shu, Jason Rambach, Alain Pagani, and Didier Stricker. Planerecnet: Multi-task learning with cross-task consistency for piece-wise plane detection and reconstruction from a single rgb image. In *Proceedings of the British Machine Vision Conference*, page 239, 2021.
- [56] Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2024.
- [57] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.
- [58] Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. Mvsnet: Depth inference for unstructured multi-view stereo. In *Proceedings of the European conference on computer vision (ECCV)*, pages 767–783, 2018.
- [59] Hanqiao Ye, Yuzhou Liu, Yangdong Liu, and Shuhan Shen. Neuralplane: Structured 3d reconstruction in planar primitives with neural fields. In *The Thirteenth International Conference on Learning Representations*, 2025.
- [60] Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. Scannet++: A high-fidelity dataset of 3d indoor scenes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12–22, 2023.
- [61] Zehao Yu, Lei Jin, and Shenghua Gao. P 2 net: Patch-match and plane-regularization for unsupervised indoor depth estimation. In *European conference on computer vision*, pages 206–222. Springer, 2020.
- [62] Zehao Yu, Songyou Peng, Michael Niemeyer, Torsten Sattler, and Andreas Geiger. Monosdf: Exploring monocular geometric cues for neural implicit surface reconstruction. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.
- [63] Zehao Yu, Torsten Sattler, and Andreas Geiger. Gaussian opacity fields: Efficient adaptive surface reconstruction in unbounded scenes. *ACM Transactions on Graphics (ToG)*, 43(6):1–13, 2024.
- [64] Farhad G Zanjani, Hong Cai, Hanno Ackermann, Leila Mirvakhabova, and Fatih Porikli. Planar gaussian splatting. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 8905–8914. IEEE, 2025.
- [65] Lantao Zhang, Haochen Niu, Peilin Liu, Fei Wen, and Rendong Ying. Efficient plane segmentation in depth image based on adaptive patch-wise region growing. *IEEE Robotics and Automation Letters*, 2025.
- [66] Wanting Zhang, Haodong Xiang, Zhichao Liao, Xiansong Lai, Xinghui Li, and Long Zeng. 2dgs-room: Seed-guided 2d gaussian splatting with geometric constraints for high-fidelity indoor scene reconstruction. *arXiv preprint arXiv:2412.03428*, 2024.
- [67] Wei Zhang, Qing Cheng, David Skuddis, Niclas Zeller, Daniel Cremers, and Norbert Haala. Hi-slam2: Geometry-aware gaussian slam for fast monocular scene reconstruction. *arXiv preprint arXiv:2411.17982*, 2024.
- [68] Zhicheng Zhang, Song Chen, Zichuan Wang, and Jufeng Yang. Planeseg: Building a plug-in for boosting planar region segmentation. *IEEE Transactions on Neural Networks and Learning Systems*, 35(8):11486–11500, 2023.
- [69] Ziyu Zhang, Binbin Huang, Hanqing Jiang, Liyang Zhou, Xiaojun Xiang, and Shunhan Shen. Quadratic gaussian splatting for efficient and detailed surface reconstruction. *arXiv preprint arXiv:2411.16392*, 2024.
- [70] Xiaowei Zhou, Haoyu Guo, Sida Peng, Yuxi Xiao, Haotong Lin, Qianqian Wang, Guofeng Zhang, and Hujun Bao. Neural 3d scene reconstruction with indoor planar priors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(9):6355–6366, 2024. doi: 10.1109/TPAMI.2024.3379833.
- [71] Matthias Zwicker, Hanspeter Pfister, Jeroen Van Baar, and Markus Gross. Ewa volume splatting. In *Proceedings Visualization, 2001. VIS'01.*, pages 29–538. IEEE, 2001.

A Technical Appendices and Supplementary Material

A.1 Additional Implementation Details

Datasets: We conduct experiments with multi-view images from three indoor datasets. For the Replica dataset [44], we use eight scenes: office0-office4 and room0-room2, 100 views sampled from each scene. For ScanNet++ [60], we select sequences 8b5caf3398, b20a261fdf, 66c98f4a9b and 88cf747085 captured by a DSLR. Our experimental data on the MuSHRoom [40] consists of five short image sequences (coffee_room, classroom, honka, kokko, and vr_room) captured by an iPhone.

Experiment Configuration: We employ an NVIDIA A6000 GPU with 48GB of VRAM for DUS3R [50], which ensures each input group to DUS3R contains 40 images, producing depth maps with better multi-view consistency. For the training process of all scenes, the depth normal consistency constraint L_{dn} is introduced from 7000 iterations, the co-planarity constraint L_p from 14000 iterations, while the prior depth and normal constraint L_{rd}, L_{rn} from 7000 and 20000 iterations respectively. This training strategy ensures Gaussians are in relatively accurate positions before introducing planar priors, thereby reducing artifact generation. For novel view reconstruction, we use every eighth image from the input data as the test set, as in 3DGS.

A.2 Additional Ablation Study

We present additional visualization of the ablation study on the MuSHRoom dataset here. Fig. 7-Fig. 9 sequentially show the ablation results of the co-planarity, geometric priors, and the depth normal consistency constraints.

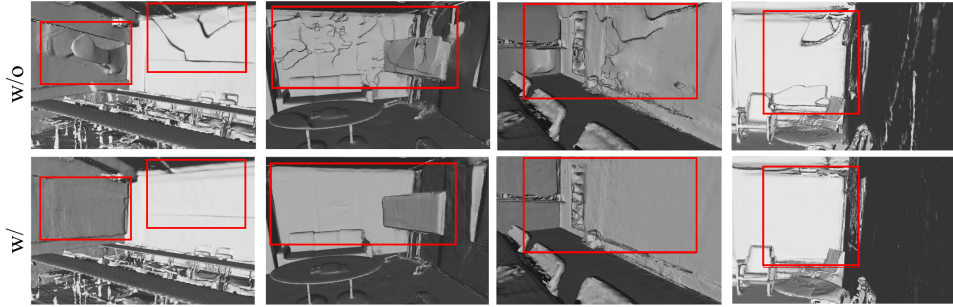


Figure 7: Ablation study of the **co-planarity constraint** on the **MuSHRoom** dataset. Improvements are highlighted in red boxes.

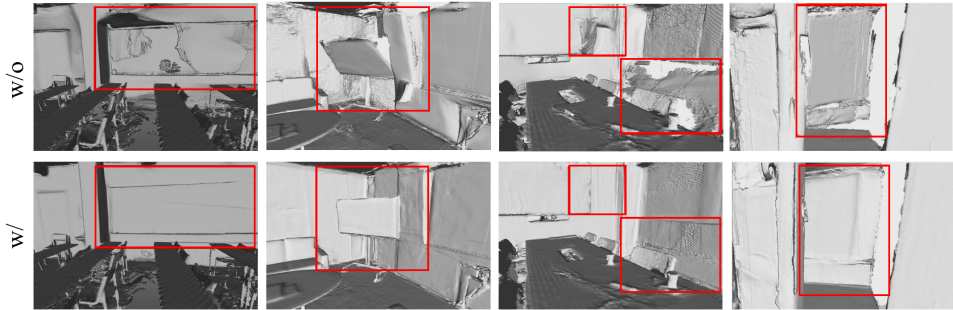


Figure 8: Ablation study of the **geometric prior constraint** on the **MuSHRoom** dataset. Improvements are highlighted in red boxes.

Co-planarity Constraint: When the co-planarity constraint is absent, Gaussian optimization relies solely on view-wise depth and normal priors, only providing local geometric guidance of the current view. The lack of global guidance introduces geometric inconsistency into the reconstructed scene, causing large planes to fragment into multiple surfaces.

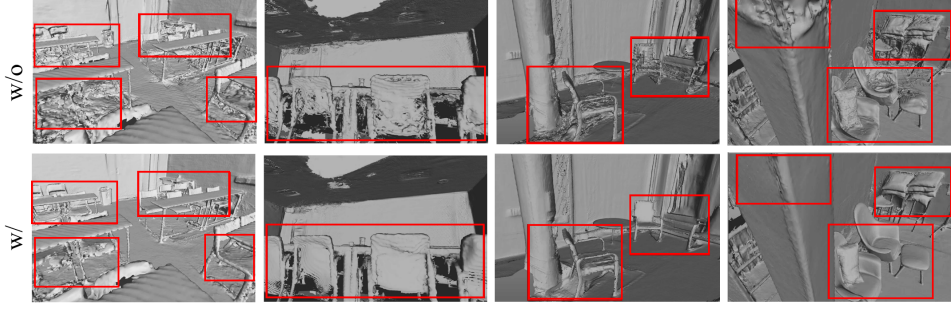


Figure 9: Ablation study of the **depth normal consistency constraint** on the **MuSHRoom** dataset. Improvements are highlighted in red boxes.

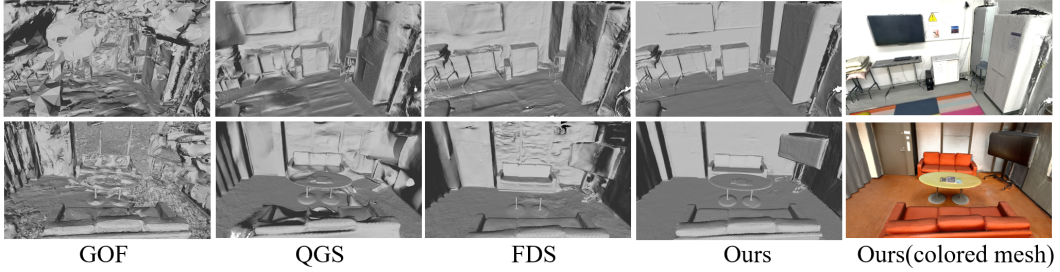


Figure 10: Additional comparison of **mesh reconstruction** on the **MuSHRoom** dataset. Compared with these recent methods, our reconstruction results are noticeably smoother in planar regions.

Geometric Prior Constraint: Without depth priors from the multi-view foundation model [50], reconstruction in texture-less regions under photometric constraint lacks essential 3D structural and scale guidance, causing significant positional inaccuracy of Gaussians. In such cases, imposing co-planarity constraints on those artifacts may generate tilted or depth-inaccurate planes.

Depth Normal Consistency: The rotation of flattened Gaussians remains unconstrained without the depth normal consistency constraint, so they tend to be perpendicular to reconstructed surfaces. This results in surface roughness similar to that observed in meshes generated by 3DGS [18].

A.3 Additional Results

Surface Reconstruction: We present a visual comparison of mesh reconstruction on the MuSHRoom dataset between our method and GOF [63], QGS [69], and FDS [7] in Fig. 10. Additionally, we present more extensive mesh reconstruction comparisons with the 3DGS [18], 2DGS [15], PGSR [4], and DN-Splatter [46] across the three datasets in Fig. 13-Fig. 15. PlanarGS achieves high-fidelity Gaussian reconstruction on all datasets, presenting significantly lower global and local errors and substantially outperforming previous methods.

Novel View Synthesis: We present the novel view synthesis results on the MuSHRoom dataset in Fig. 11, comparing our method with the ground truth, 3DGS [18], PGSR [4], and DN-Splatter [46]. It can be seen that in low-texture and reflective planar regions, other methods lack geometric constraints and produce artifacts in novel views due to the absence of planar priors, whereas our results are closer to the ground truth.

Plane Masks and Rendering Results: More comparisons with ZeroPlane [29] and GroundedSAM [39] in terms of planar-prior generation are shown in Fig. 12. We also show visualizations of planar prior masks and rendering results of PlanarGS on three datasets in Fig. 16-Fig. 18. The first row displays the RGB images rendered by PlanarGS following Eq. 3, the second row shows the plane masks obtained by our pipeline for Language-Prompted Planar Priors (see Sec. 3.2), while the third and fourth rows exhibit the normal and depth maps rendered by PlanarGS through Eq. 7 and Eq. 9, respectively. Our method achieves robust planar prior masks and accurate RGB and geometric rendering, maintaining geometric consistency without compromising the rendering and reconstruction accuracy of Gaussian splatting.

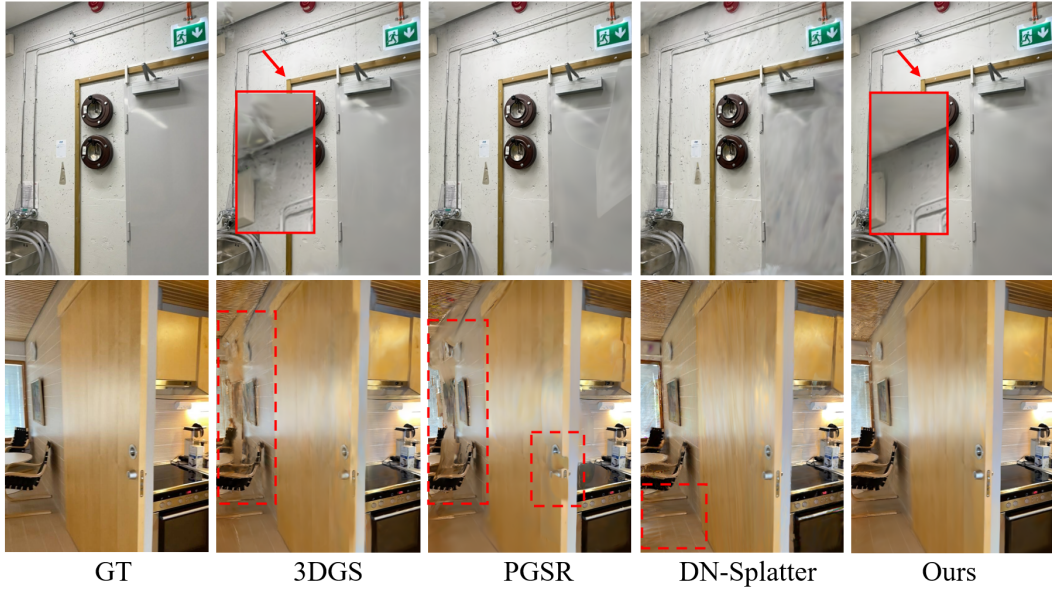


Figure 11: **Additional novel view synthesis visualization.** The highlighted regions are **enlarged** or **marked** with red boxes.



Figure 12: **Comparison of planar priors.** ZeroPlane fails to distinguish fine details or curved surfaces as non-planar regions, while GroundedSAM suffers from plane omission and merges planar regions.

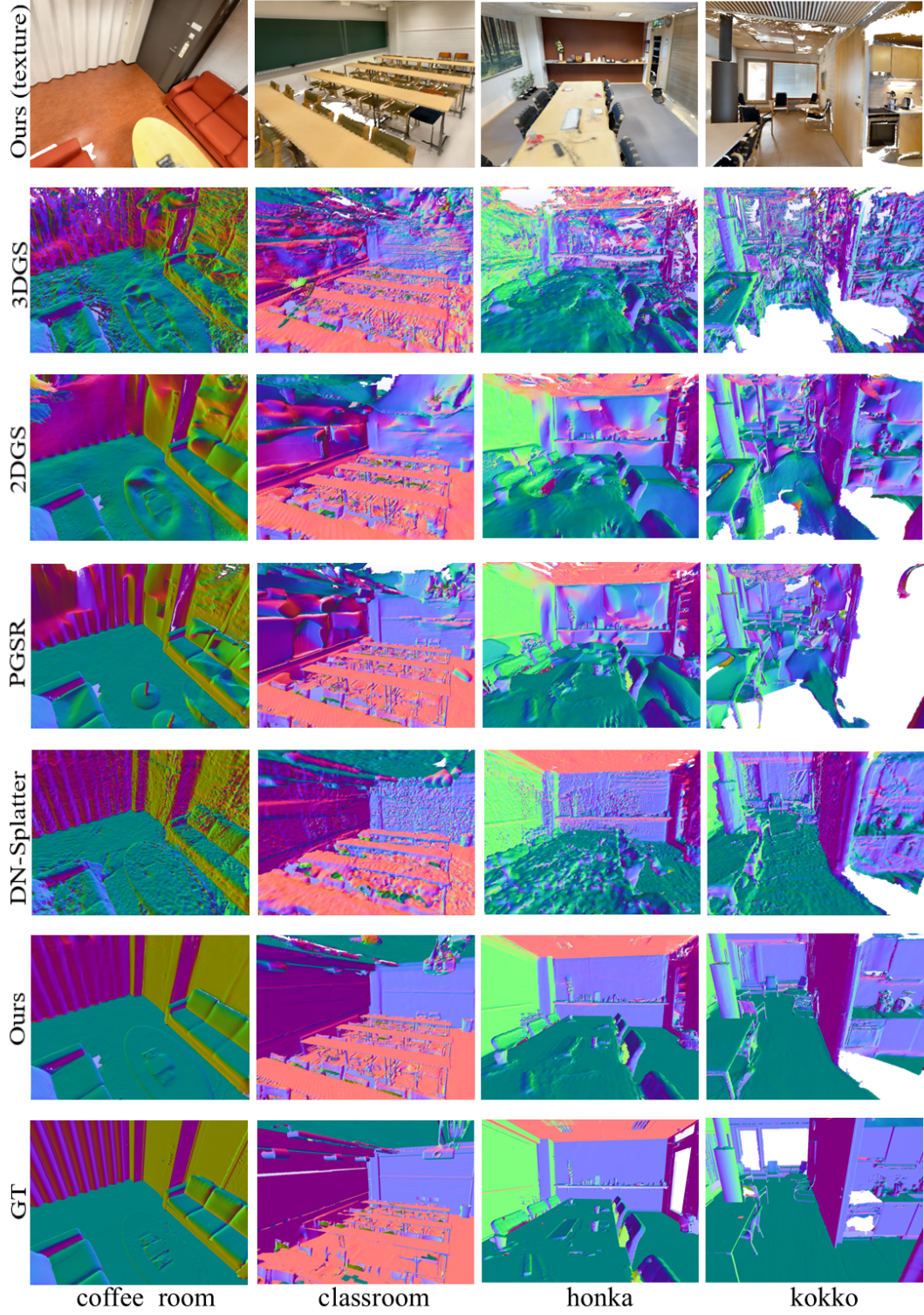


Figure 13: Additional qualitative comparison on the **MuSHRoom** dataset. We color the mesh according to surface normal directions to facilitate visual assessment of planarity consistency. Our meshes colored with textures are presented in the first row as references.

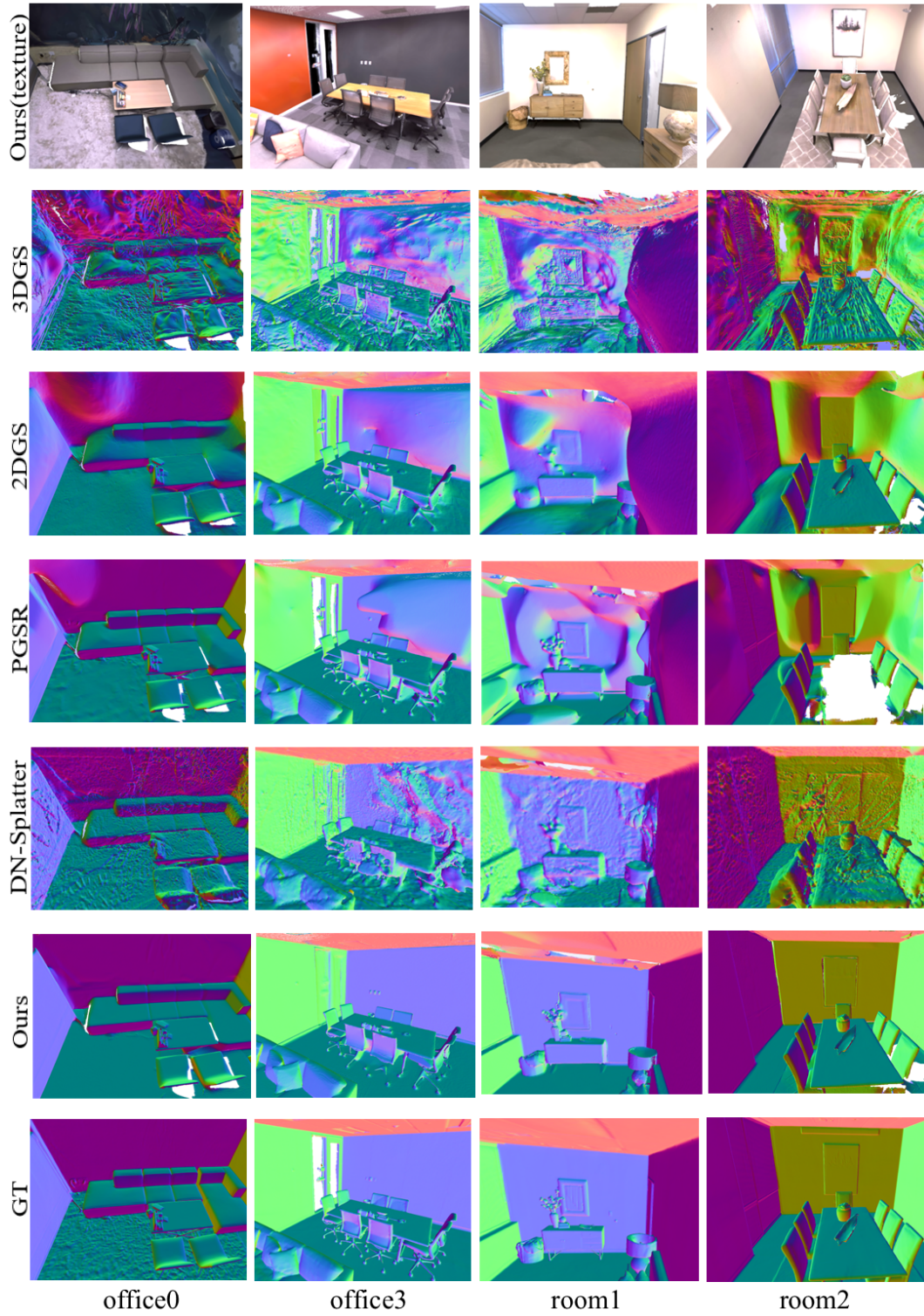


Figure 14: Additional qualitative comparison on the **Replica** dataset. We color the mesh according to surface normal directions to facilitate visual assessment of planarity consistency. Our meshes colored with textures are presented in the first row as references.

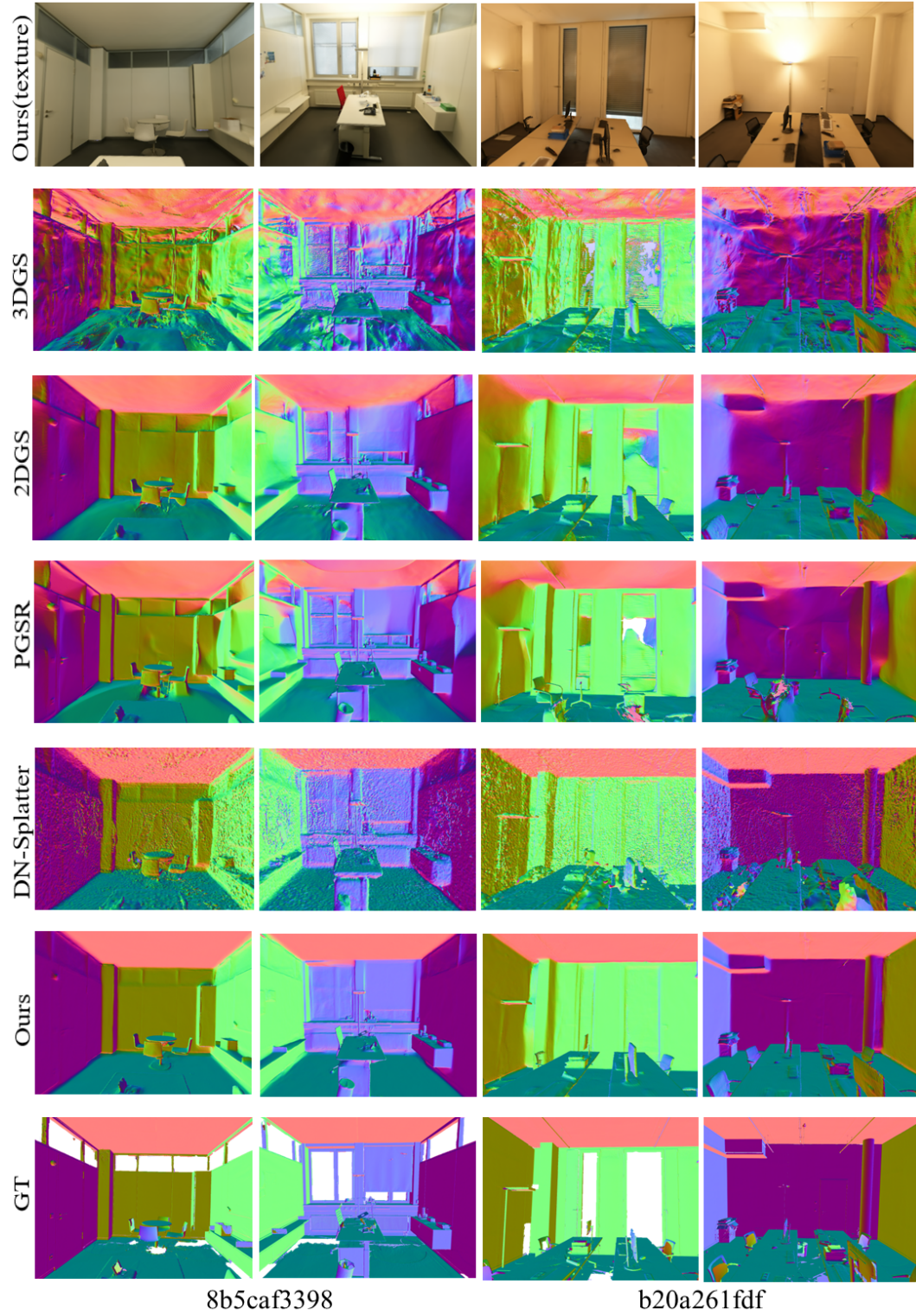


Figure 15: Additional qualitative comparison on the **ScanNet++** dataset. We color the mesh according to surface normal directions to facilitate visual assessment of planarity consistency. Our meshes colored with textures are presented in the first row as references.

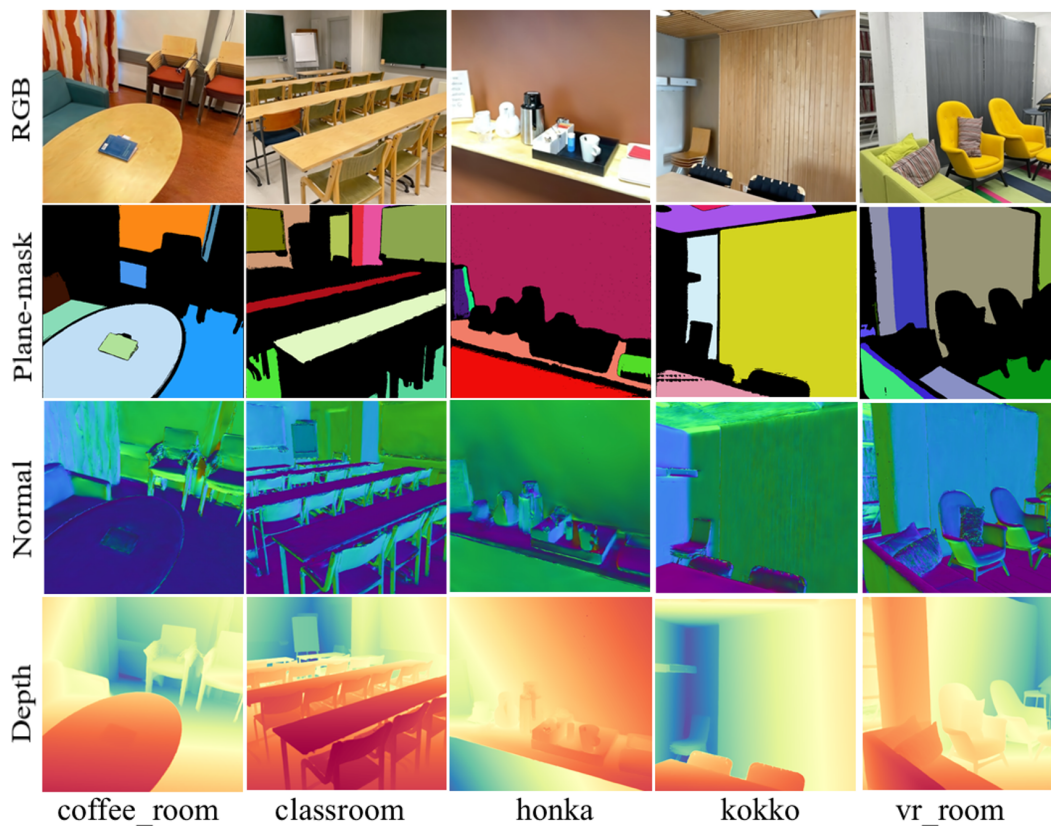


Figure 16: Rendering and plane-mask results on the **MuSHRoom** dataset.

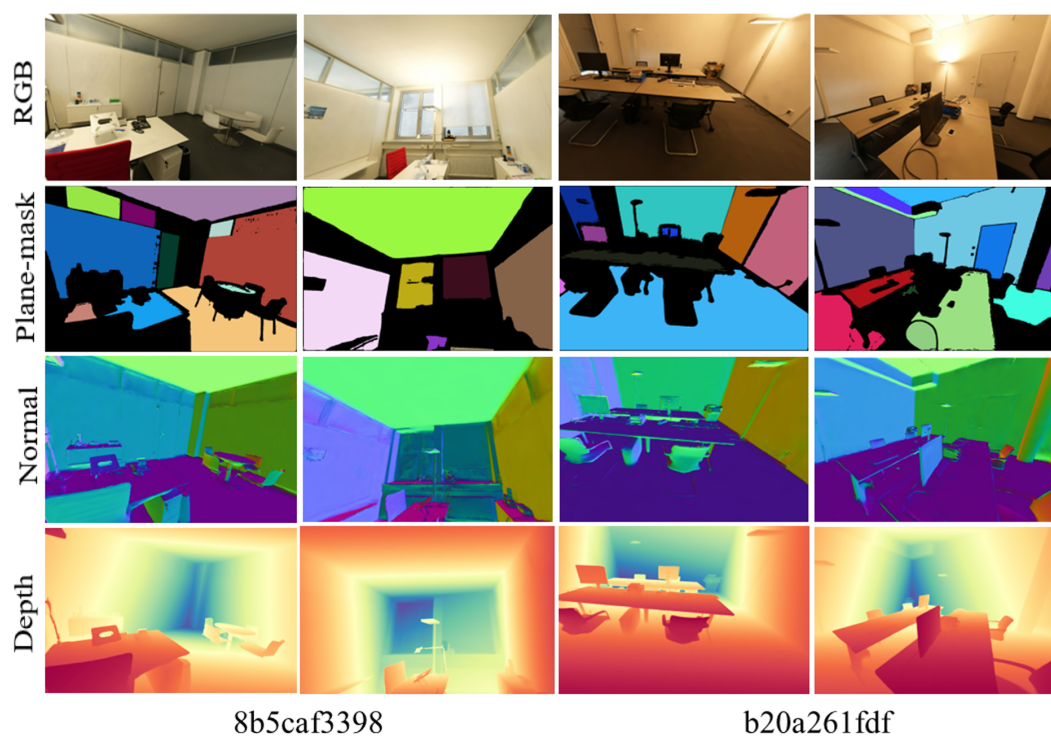


Figure 17: Rendering and plane-mask results on the **ScanNet++** dataset.

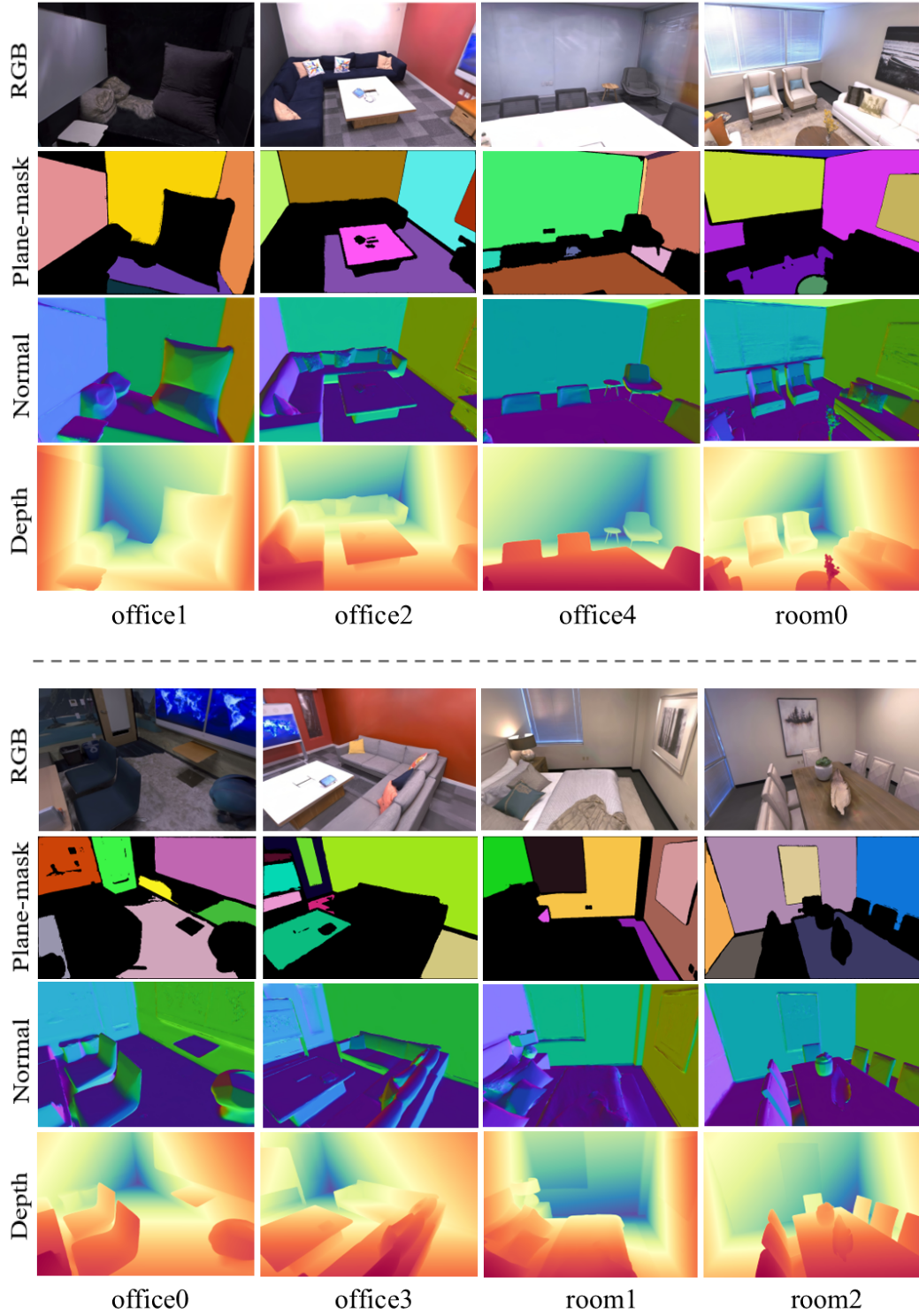


Figure 18: Rendering and plane-mask results on the **Replica** dataset.

NeurIPS Paper Checklist

1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

Answer: [\[Yes\]](#)

Justification: We claim to obtain high-fidelity image-based 3D reconstruction on indoor scenes based on 3D Gaussian Splatting. Our contributions can be found both in the abstract and at the end of the introduction. Our claims do match our theoretical and experimental results.

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [\[Yes\]](#)

Justification: We discuss our limitations of application scenes in the conclusion section.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. Theory assumptions and proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [\[Yes\]](#)

Justification: We provide sufficient formulas for our theories and assumptions in the method section that are numbered, cross-referenced and properly referenced.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. Experimental result reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [\[Yes\]](#)

Justification: We have fully described our novel architecture in the method section and claimed the datasets we used.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
 - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
 - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
 - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
 - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: Our code haven't been released.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. Experimental setting/details

Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: We specify our implementation details in the experiments section and more details are in the supplementary materials.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. Experiment statistical significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: We follow the existing related works on the statistical significance of the experiments.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. Experiments compute resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: We provide our type of computer workers and time of execution.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. Code of ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines>?

Answer: [Yes]

Justification: We obey the NeurIPS Code of Ethics and keep anonymity.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. Broader impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [NA]

Justification: Our method is about improving the performance of 3D reconstruction from multi-view images, which does not have more societal impacts than existing methods.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: Our paper pose no such risks.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We have cited the original paper that produced the code package or dataset in the experiment section.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA]

Justification: The paper does not release new assets.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: We does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (<https://neurips.cc/Conferences/2025/LLM>) for what should or should not be described.