Structure Gaussian Splatting SLAM

Yan Li[1], Yingzhao Li[2], and Gim Hee Lee[1]

[1] National University of Singapore, Singapore
[2] Harbin Institute of Technology, China

Abstract. Recent Gaussian Splatting SLAM (GS-SLAM) methods align rendered and observed images through photometric and geometric residuals to achieve high-quality reconstruction and rendering. However, these signals weaken in low-texture and repetitive environments, which leads to floaters, inconsistent geometry, and drift. Existing systems mitigate drift through loop closure or global layout assumptions, but these mechanisms are opportunistic or lack persistent structural entities that provide reliable constraints throughout mapping. We introduce a structure-aware GS-SLAM framework that models planar structures as persistent *Planar Gaussian Instances* (PGIs) within a 3D Gaussian map. Each PGI represents a planar surface associated with Gaussian primitives and maintains a consistent identity across frames. This representation promotes planes from transient observations to map entities and converts planar structures into stable geometric constraints. Building on PGIs, we propose *structure closure*, which estimates relative camera pose from multiple non-parallel planes without requiring trajectory revisits. We further integrate PGIs into a joint optimization that combines photometric–depth supervision with plane-instance consistency to improve multi-view geometric coherence and suppress floater growth. Experiments on public benchmarks show that the proposed system achieves state-of-the-art performance in camera tracking, dense reconstruction, and novel-view rendering, with strong robustness in scenes with large textureless surfaces and repetitive layouts. Our code is publicly available at https://github.com/yanyan-li/StructureGS-SLAM.

Keywords: Gaussian Splatting SLAM · Planar Structure · Structure Closure

1 Introduction

Modern AR/VR, mobile robotics, and digital twins require dense and photorealistic 3D maps for precise localization and structure-aware reasoning. RGB-D video provides a natural sensing modality. Depth stabilizes geometry in texture-poor regions, while RGB preserves appearance and semantics. Among available representations, 3D Gaussian Splatting (3DGS) [5] is particularly attractive because it supports efficient rendering and offers a compact differentiable model of scene geometry and appearance. Recent work integrates Gaussian splatting into SLAM systems to jointly estimate camera poses and dense Gaussian

Fig. 1: Example results from the proposed Structure Gaussian Splatting SLAM system. The left part visualizes the reconstructed 3D scene in depth, plane-instance, and appearance formats; the top-right shows novel-view synthesis during tracking; and the bottom-right displays the reconstructed 3D Gaussian primitives of the scene.

maps [4, 14, 31]. These systems supervise tracking and reconstruction through photometric and geometric consistency between rendered views and RGB-D observations. However, such signals weaken in low-texture surfaces, specular regions, and repetitive indoor layouts. As a result, drift can accumulate even when dense depth measurements are available.

Existing systems address long-term drift through loop closure strategies inherited from classical SLAM. Some methods maintain Gaussian submaps and perform global alignment across revisited regions [34,36]. Others rely on learned global descriptors or feature correspondences to detect revisits [12,18]. Loop closure improves global consistency when the trajectory revisits previously observed locations with sufficient visual overlap. However, such revisits are often rare or ambiguous in corridors and repetitive rooms. Structural priors such as Manhattan or Atlanta world assumptions impose global orthogonality constraints between dominant planes [8, 19, 32]. Although effective in structured scenes, these assumptions are brittle when environments deviate from idealized layouts. Consequently, existing Gaussian SLAM systems rely either on opportunistic loop closure or restrictive global assumptions, and seldom maintain persistent structural entities that can be matched and optimized reliably across time.

Fortunately, indoor environments contain abundant planar structures such as walls, floors, tables, and cabinets. These planes remain stable across viewpoints and impose strong geometric constraints on camera motion. When represented explicitly, they can provide frequent and reliable constraints that do not depend on trajectory revisits or global Manhattan assumptions. Motivated by this observation, we introduce a structure-aware Gaussian SLAM framework that models planes as persistent *Planar Gaussian Instances* (PGIs) within a 3D Gaussian map, as illustrated in Fig. 1. Each PGI represents a continuous planar surface associated with a set of Gaussians and maintains a consistent identity across frames. Unlike existing GS-SLAM systems that treat planar observations as transient cues during tracking or reconstruction, our approach promotes planes to persistent map entities. This representation keeps planar structures identifiable throughout mapping and converts planar observations into explicit geometric constraints across time.

Building on PGIs, our system maintains consistent plane identities through plane detection, patch merging, and multi-view association. These persistent planar entities enable a new geometric constraint that we call *structure closure*. For two frames that observe two or more non-parallel planes, we estimate the relative pose from plane normals and offsets and add the resulting constraint to the pose graph alongside standard photometric and geometric supervision. Unlike loop closure, structure closures do not require revisiting the same location. They therefore provide frequent opportunities to correct drift in long corridors and repetitive indoor environments.

We further integrate PGIs into a joint optimization that combines Gaussian photometric-depth consistency with plane instance regularization. This formulation strengthens supervision in low-texture regions and suppresses the growth of spurious Gaussians that appear as floaters in GS-SLAM maps. Sparse global viewpoints and bundle adjustment over camera poses and Gaussians maintain long-range consistency. Our design converts ubiquitous planar structures into reliable geometric constraints, which correct drift earlier and more frequently than loop-closure-based approaches while preserving the efficiency and visual quality of Gaussian splatting. This improves robustness for RGB-D mapping in indoor environments with large textureless surfaces and repeated layouts. Our main contributions are summarized as follows:

- **PGI representation.** We introduce a structure-aware Gaussian map that augments Gaussian primitives with persistent planar identities. The representation detects planar regions, merges fragmented patches, and maintains consistent plane instances across frames through multi-view association.
- **Structure closures for GS-SLAM.** We derive relative pose constraints from observations of two or more non-parallel planes and incorporate them into a pose graph. These constraints correct drift without requiring loop revisits or Manhattan-world assumptions.
- **Structure-aware joint optimization.** We combine low-texture enhanced photometric and depth supervision with plane instance consistency to stabilize long-horizon mapping. The optimization jointly refines Gaussian parameters and camera poses under both appearance and structural constraints.
- **Comprehensive evaluation.** Extensive experiments demonstrate improved accuracy and robustness compared with state-of-the-art Gaussian SLAM methods across challenging indoor environments.

2 Related Work

Classical SLAM. Classical visual SLAM estimates 6-DoF camera poses and incrementally reconstructs scenes from monocular, stereo, or RGB-D streams [15, 16, 27]. In man-made environments, low texture and repeated layouts weaken keypoint matches and induce drift. Prior work addresses these degeneracies through several directions: i) multi-type geometric features such as points, lines, and planes [1, 13, 25]; ii) structural priors such as Manhattan or Atlanta worlds that impose dominant orientation constraints [8, 19, 32]; iii) learning-based priors that predict depth, normals, or optical flow to improve tracking robustness [9, 22, 23, 28]; and iv) loop closure that corrects drift when revisiting previously observed locations [6, 16]. However, these strategies have inherent limitations. Global frame assumptions can fail in general layouts, learned priors improve local alignment but do not enforce structural consistency across views, and loop closure remains opportunistic because it requires strong visual overlap. In contrast, our method derives *structure closures* from matched non-parallel planes to provide frequent geometry-grounded pose constraints even in low-texture and repetitive environments. By representing planes as persistent *Planar Gaussian Instances (PGIs)*, the system maintains cross-view structural consistency directly in the map without assuming Manhattan or Atlanta alignment.

Gaussian Splatting SLAM. Following the success of 3D Gaussian Splatting [5], a growing body of work has incorporated Gaussian representations [3, 5, 10] into SLAM to jointly optimize camera poses, dense scene reconstruction, and novel view synthesis. MonoGS [14] pioneered online incremental Gaussian mapping with adaptive density control. SplaTAM [4] improves mapping efficiency by representing scenes with smooth 3D Gaussians, while Gaussian-SLAM [31] adopts a submap-based representation for scalable optimization [17, 33]. Recent advances further extend GS-SLAM toward uncertainty-aware optimization (CG-SLAM [2], VarSplat [24]), semantic scene understanding (SGS-SLAM [7]), dynamic scene modeling, and efficient large-scale reconstruction, substantially improving the robustness and applicability of Gaussian-based SLAM systems. To improve global consistency, LoopSplat detects loop closures between Gaussian submaps [34], and Loopy-SLAM introduces loop edges for neural point maps [12, 18]. However, these pipelines still rely primarily on visual loop closure and do not explicitly maintain planar structures as persistent optimizable entities. Compared to existing GS-SLAM methods, our approach preserves the visual fidelity of Gaussian splatting while explicitly modeling large surfaces as persistent *Planar Gaussian Instances (PGIs)* and forming *structure closures* from matched planes. These constraints allow drift to be corrected earlier and more frequently without requiring image-level revisits. Additionally, we introduce texture-aware RGB-Depth consistency constraints that mitigate the gradient-vanishing issue of traditional photometric losses and improve tracking robustness in texture-poor regions.

3 Our Methodology

Overview. Fig. 2 illustrates our structure-aware GS-SLAM system, which builds a 3D Gaussian map from an RGB-D sequence by promoting planes to persistent entities and converting planar matches into pose constraints. We represent the scene using *Planar Gaussian Instances* (PGIs), which expose mid-level structure directly in the map. The front-end tracks the camera and selects keyframes using

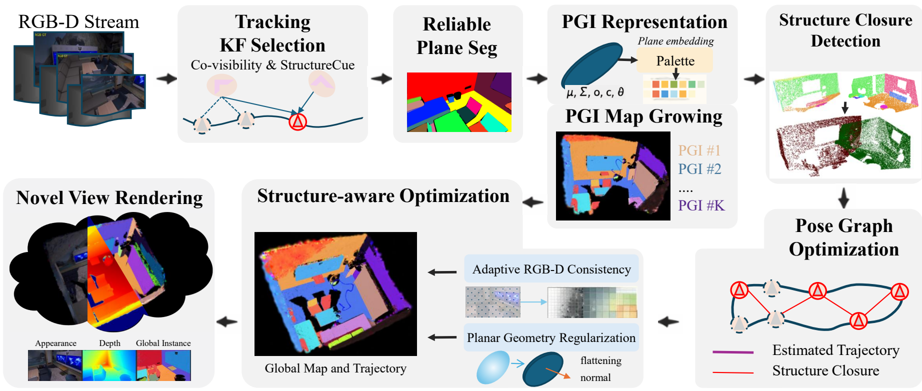


Fig. 2: Overview of our system. From an RGB-D stream, we select keyframes (KF) using co-visibility and structural cues, extract per-keyframe plane segments, and associate them with global planar instances (PGIs). Consistent planes across non-adjacent frames trigger geometry-derived constraints, *structure-closures*, which do not require image-level revisits that link new keyframes to anchor viewpoints. A sliding-window optimizer then jointly refines active poses and Gaussian parameters with anchors stabilizing long-horizon drift.

both co-visibility and structural cues (Sec. 3.1). For each keyframe, we extract reliable planes and associate them with global PGIs through geometry-based matching in the world frame followed by appearance-guided refinement using rendered PGI and RGB maps (Sec. 3.2). Next, we establish *structure closures* between non-adjacent frames by matching two or more non-parallel planes and refining the estimate with sparse non-planar correspondences (Sec. 3.3). These constraints are added to a pose graph for drift correction. Finally, a sliding-window optimizer jointly refines active camera poses and Gaussian parameters, while a small set of *Sparse Global Viewpoints* (SGVs) remains fixed to preserve long-horizon consistency (Sec. 3.4).

3D Gaussian Representation. Each Planar Gaussian Instance is modeled with a set of modified Gaussian primitives, $\mathcal{G}_i = [\boldsymbol{\mu}_i,\ \boldsymbol{\Sigma}_i,\ o_i, \boldsymbol{c}_i,\ \boldsymbol{\theta}_i]$, where $\boldsymbol{\mu}_i \in \mathbb{R}^{3\times1}$ is the mean, $\boldsymbol{\Sigma}_i \in \mathbb{R}^{3\times3}$ is the covariance, o_i is the opacity, $\boldsymbol{c}_i$ is the color parameter, and $\boldsymbol{\theta}_i \in \mathbb{R}^{3\times1}$ is a plane instance embedding. A set of co–planar Gaussians with a common global plane identity forms a PGI. Besides RGB and depth, we render an instance map with standard rasterization and alpha blending to supervise $\boldsymbol{\theta}_i$. Refer to the Supplementary Material for the rasterization and alpha blending processes.

3.1 The Front-End

The front-end estimates camera poses and selects informative keyframes while explicitly leveraging planar structure exposed by PGIs. Unlike prior GS-SLAM systems that rely primarily on photometric alignment and view overlap, our front-end introduces structure-aware tracking and keyframe selection to improve robustness in low-texture and repetitive environments.

Camera Tracking. For each new RGB-D frame, we estimate the camera pose $\mathbf{T}_{c,w} \in \mathrm{SE}(3)$ by aligning differentiable renders of the current Gaussian map $\mathcal{M}$ with observations. Conventional GS-SLAM methods rely mainly on an L_1 photometric loss [4,14], which becomes unstable in texture-poor regions. We instead employ an *adaptive RGB-Depth consistency loss* that jointly enforces photometric alignment, depth agreement, and texture-aware structural cues. Specifically, Sobel-based texture detection identifies informative regions, while gradient and SSIM terms strengthen structural alignment. Charbonnier penalties further improve robustness to depth noise and specularities. This design provides significantly more stable pose estimation in large planar or low-texture scenes.

Keyframe Selection. We insert keyframes using two complementary signals that capture both appearance coverage and structural change.

1) Co-visibility. Following MonoGS [14], we measure the overlap between the current frame and the last keyframe by counting Gaussians that are jointly visible under rasterization or accumulated ray opacity. Low overlap indicates new scene coverage and triggers keyframe insertion.

2) Structure cue. Beyond appearance overlap, we detect geometric change in dominant planar structures. For each detected plane we maintain its world-frame parameters $(\mathbf{n}_w, d_w)$ and its camera-frame parameters at the last keyframe $(\mathbf{n}_c^{\text{last}}, d_c^{\text{last}})$. Given the current pose, we transform $(\mathbf{n}_w, d_w)$ into the current camera frame to obtain $(\mathbf{n}_c^{\text{cur}}, d_c^{\text{cur}})$. A plane is considered drifted if $\angle(\mathbf{n}_c^{\text{cur}}, \mathbf{n}_c^{\text{last}}) > \tau_\theta$ or $|d_c^{\text{cur}} - d_c^{\text{last}}| > \tau_d$. We insert a keyframe when the number of drifted planes exceeds $\max(1, \tau_{\text{ratio}} \cdot N)$, where N is the number of observed planes. This structure cue captures viewpoint changes such as turning a corner or revealing new planar surfaces, where co-visibility alone may remain high.

We insert a keyframe when either signal is triggered. Appearance and structure cues ensure that keyframes capture both novel geometry and informative structural viewpoints.

3.2 From 2D Plane Segments to Planar Gaussians

Single-View Plane Instances. We extract candidate planar structures from each newly inserted keyframe to initialize potential PGIs. Starting from the RGB-D image, we detect planar regions using a graph-based segmenter [11,26] and retain the K largest segments by pixel area (default $K = 10$) to focus on dominant scene surfaces. For each retained segment, pixels are back-projected into 3D and a plane is fitted in the current camera frame to obtain $\pi_c = [\,\mathbf{n}_c, d_c\,]$ as defined in Sec. 3.1. Segment labels are then propagated to the down-sampled point cloud so that subsequent modules operate directly on 3D supports.

Single-view plane extraction is often unreliable due to two common failure modes: 1) ***Over-segmentation*** occurs when a single physical plane appears as

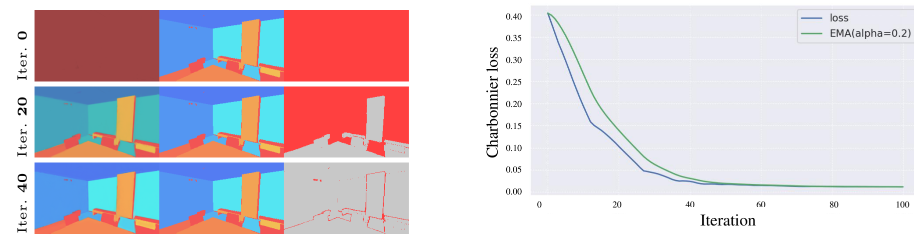


Fig. 3: Planar instance learning in initialization during the optimization process (Iter. $0 \rightarrow 40$). The left block shows rendered plane segments across iterations, while the right block shows the corresponding ground-truth results and errors. Loss curves of optimization based on the proposed plane embedding.

multiple disconnected patches due to occlusions or texture changes. 2) ***Low-quality extraction*** arises when non-planar clutter is mistakenly labeled as planar. Such errors are particularly harmful for PGI construction because they introduce fragmented or unstable planar entities. To address these issues, we introduce a lightweight refinement stage tailored for PGI initialization. Over-segmented planes are consolidated by merging segments that agree in plane parameters and exhibit consistent spatial support, followed by a refit on the merged points. Low-quality segments are filtered using planarity support statistics and the stability of the fitted plane inherited from the structure-cue step, which avoids redundant thresholding while preserving reliable planes.

Importantly, non-planar points are retained to provide sparse geometric correspondences during structure-closure estimation. Ambiguous planar segments are also preserved and deferred to the local–global PGI association and multi-view relaxation stages, where additional viewpoints help resolve segmentation errors and enforce cross-view consistency.

3.3 Global Planar Gaussians

We maintain persistent planar structures as PGIs that evolve across keyframes. Starting from the first keyframe, planar segments are lifted into the global Gaussian map and subsequently associated, validated, and expanded as new observations arrive. This design transforms transient 2D plane detections into persistent map entities that accumulate geometric support over time.

Planar Gaussian Initialization. For the first keyframe c_0, we take its camera frame as the canonical world frame and extract labeled 3D point sets for each detected planar segment. From these points we estimate plane parameters $\pi_{c_0}^k = [\,\mathbf{n}_{c_0}^k,\ d_{c_0}^k\,]$, where $\mathbf{n}_{c_0}^k \in \mathbb{S}^2$ and $d_{c_0}^k \in \mathbb{R}$. The Gaussian map is then initialized with both planar and non-planar regions. Beyond the standard RGB and depth fields, we introduce a *plane-instance rendering channel* that supervises the identity of planar Gaussians. Specifically, each Gaussian stores a continuous embedding:

$$\boldsymbol{\theta}_i = (\theta_i^R, \theta_i^G, \theta_i^B), \qquad \mathbf{c}_i = \sigma(\boldsymbol{\theta}_i) \in (0,1)^3, \tag{1}$$

where $\sigma(\cdot)$ is the sigmoid function. The ground-truth plane mask $id(\cdot)$ at pixel $\mathbf{u}$ is mapped to a stable color palette:

$$\mathbf{y}(\mathbf{u}) = \text{Palette}(\text{id}(\mathbf{u})) \in [0, 1]^3, \tag{2}$$

and the rendered prediction of the Gaussian map is:

$$\hat{\mathbf{c}}(\mathbf{u}) = (\hat{c}^R(\mathbf{u}), \hat{c}^G(\mathbf{u}), \hat{c}^B(\mathbf{u})). \tag{3}$$

We supervise this plane-instance field using an edge-aware Charbonnier loss:

$$E_{\text{pgi}} = \frac{1}{|\Omega|} \sum_{u \in \Omega} w_{\text{edge}}(u) \sqrt{\|\hat{\mathbf{c}}(u) - \mathbf{y}(u)\|_2^2 + \varepsilon^2}, \tag{4}$$

where $w_{\text{edge}}(u)$ emphasizes plane boundaries using image gradients. This continuous embedding allows planar identities to be optimized jointly with Gaussian parameters while preserving sharp instance boundaries, as shown in Fig. 3.

Planar Gaussian Growing. For each subsequent keyframe c_i, reliable single-view planes $\pi_{c_i} = [\mathbf{n}_i, d_i]$ are extracted in the camera frame and transformed into the world frame using the current pose estimate $\mathbf{T}_{c_i,w}$. We then associate newly observed planar regions with existing PGIs through a two-stage matching process: **1)** ***Geometric gating.*** For every global plane $P_g = (\mathbf{n}_g, d_g)$ we compute the angular and offset differences:

$$\theta_{ig} = \arccos(\mathbf{n}_i^\top \mathbf{n}_g), \qquad \Delta d_{ig} = \frac{|d_i - d_g|}{\max(|d_i|, |d_g|, \varepsilon)}. \tag{5}$$

Pairs exceeding thresholds $(\theta_{\max}, \delta_{\max})$ are rejected. **2)** ***Projected overlap.*** The remaining candidates are validated by projecting the current plane points $\mathcal{X}_i$ and sampled global plane points P_g to the image plane to form binary masks $\mathbf{M}_i$ and $\mathbf{M}_g$. Their overlap is measured as:

$$\text{IoU}(i, g) = \frac{|\mathbf{M}_i \cap \mathbf{M}_g|}{|\mathbf{M}_i \cup \mathbf{M}_g|}. \tag{6}$$

For valid pairs we define a matching cost:

$$c_{ig} = w_\theta \frac{\theta_{ig}}{\theta_{\max}} + w_d \frac{\Delta d_{ig}}{\delta_{\max}} + w_o(1 - \text{IoU}(i, g)), \tag{7}$$

and solve a one-to-one assignment with the Hungarian algorithm. w_θ, w_d, w_o are weights to balance the cost terms. Unmatched planes are instantiated as new PGIs and their Gaussians are appended to the global map, which allow planar structures to grow incrementally as the camera explores the scene.

3.4 Structure-aware Global Optimization

We jointly optimize camera poses and 3D Gaussian primitives while explicitly enforcing planar structure through PGIs. Our formulation integrates four complementary mechanisms: 1) ***structure-closure constraints*** derived from matched non-parallel planes, 2) ***Sparse Global Viewpoints*** (SGVs) that anchor the optimization to stabilize long-horizon trajectories, 3) ***planar geometry regularization*** that maintains coherent Gaussian representations on planar surfaces, and 4) ***adaptive RGB-Depth consistency*** that balances photometric and geometric supervision under varying texture conditions.

Pose Graph Construction. We maintain a pose graph $G = (V, E)$ over selected keyframes, where each vertex $i \in V$ stores the camera pose $\mathbf{T}_{c_i,w} \in SE(3)$. Edges encode two types of relative pose constraints: 1) ***Odometry edges*** connect temporal neighbors using: $\hat{\mathbf{T}}_{c_{i+1},c_i} = \mathbf{T}_{c_{i+1},w}\mathbf{T}_{c_i,w}^T$. 2) ***Structure-closure edges*** connect non-adjacent frames that co-observe at least two matched PGIs with non-parallel normals. Plane normals determine the relative rotation while offsets provide linear constraints on translation to give an initial relative pose estimate. To improve robustness under noise, we refine this estimate using a lightweight multi-plane Iterative Closest Point (ICP) registration between the corresponding planar point sets. The refined transformation is then inserted into the pose graph as a structure-closure constraint. Unlike loop closure, these geometry-derived constraints do not require visual overlap and therefore provide frequent drift correction in low-texture and repetitive indoor environments.

Sparse Global Viewpoints (SGVs). Sliding-window optimization over only recent frames tends to accumulate long-horizon drift [14], while optimizing the entire trajectory is computationally expensive. To stabilize the global reference frame, we maintain a small set of anchor keyframes called ***Sparse Global Viewpoints*** (SGVs) whose poses remain fixed during optimization. SGVs are selected from outside the active window according to structural connectivity: 1) keyframes participating in structure-closure edges, 2) keyframes sharing at least two PGIs with the active window, and 3) highly connected nodes in the pose graph as fallback. These anchors provide global structural constraints that stabilize the optimization without requiring full-history updates.

Planar Geometry Regularization. To ensure that Gaussians associated with PGIs remain geometrically consistent with planar surfaces, we introduce two lightweight regularizers: *flattening* and *normal alignment*. Flattening encourages the Gaussian ellipsoid to become thin along the plane normal direction:

$$L_{\text{flat}} = \left(s_3 - k\sqrt{s_1 s_2}\right)^2, \tag{8}$$

where (s_1, s_2, s_3) denote the ordered Gaussian scales and k controls planar thinness. Normal alignment further encourages the minor axis of each Gaussian to align with the corresponding plane normal:

$$L_{\text{n}} = \mathbb{E}_i\left[1 - |\mathbf{n}_g \cdot \mathbf{n}_p|\right]. \tag{9}$$

These regularizers work together to suppress spurious volumetric Gaussians and promote thin surfel-like representations on planar surfaces. The two losses are weighted by ω_{n} and ω_{flat} in the joint optimization objective: $E_{str} = \omega_{\mathrm{n}} L_{\mathrm{n}} + \omega_{\mathrm{flat}} L_{\mathrm{flat}}$.

Adaptive RGB-Depth Consistency. Standard photometric losses become unreliable in texture-poor regions where image gradients vanish. We therefore introduce a texture-aware hybrid loss that adaptively balances photometric and geometric supervision. A Sobel operator estimates texture strength:

$$G = \sqrt{(\partial_x I_{gt})^2 + (\partial_y I_{gt})^2}, \tag{10}$$

from which we derive a normalized gate:

$$\mathcal{T}(u, v) = 1 - \frac{G(u, v) - G_{\min}}{G_{\max} - G_{\min} + \epsilon}. \tag{11}$$

Low-texture pixels receive stronger geometric supervision, while high-texture regions prioritize photometric alignment. Photometric residuals combine intensity, gradient, and SSIM terms:

$$\begin{aligned} E_{\mathrm{pho}} = {} & w_{\mathrm{L1}} \|I_r - I_{gt}\|_1^{M_{\mathrm{rgb}}} + w_{\mathrm{grad}} \rho(\nabla I_r - \nabla I_{gt})^{\mathcal{G} M_{\mathrm{rgb}}} \\ & + w_{\mathrm{ssim}} (1 - \mathrm{SSIM}(I_r, I_{gt}))^{\mathcal{G} M_{\mathrm{rgb}}}, \end{aligned} \tag{12}$$

while depth alignment uses a Charbonnier penalty:

$$E_{\mathrm{geo}} = \rho(D_r - D_{gt})^{M_d}, \tag{13}$$

which provides stable gradients and reduces the influence of large residuals.

Total Objective. We optimize active-window poses and Gaussian parameters with SGVs fixed by minimizing:

$$\mathcal{L} = \lambda_{\mathrm{pgi}} E_{\mathrm{pgi}} + \lambda_{\mathrm{pho}} E_{\mathrm{pho}} + \lambda_{\mathrm{geo}} E_{\mathrm{geo}} + \lambda_{\mathrm{str}} E_{\mathrm{str}}, \tag{14}$$

where $\lambda_{\mathrm{pgi}}, \lambda_{\mathrm{pho}}, \lambda_{\mathrm{geo}}$, and λ_{str} balance structural consistency against image-, depth-, and structure-space fidelity. This objective preserves the rendering efficiency of Gaussian splatting while injecting strong planar structural constraints into the optimization.

4 Experiment

4.1 Experimental Setup

Implementation and Baselines. We evaluate our proposed SLAM on a workstation equipped with an Intel Core i9 CPU and a single NVIDIA GeForce RTX 4090 GPU. Following the design of 3DGS and Gaussian Splatting SLAM system, all time-critical rasterization and gradient computations are implemented in CUDA, while the remainder of the SLAM pipeline is built using PyTorch. Detailed hyperparameter configurations are provided in the Supplementary Material. For performance comparison, we select two categories of state-of-the-art baselines: neural-implicit methods and Gaussian-Splatting approaches.

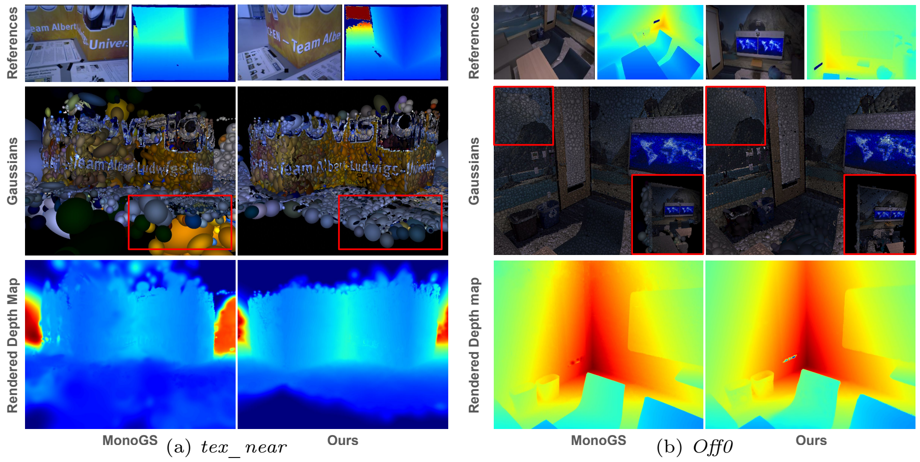


Fig. 4: Qualitative comparison in 3D Gaussian primitives and depth rendering in Replica and TUM RGB-D datasets. More qualitative results are added to the Supplementary Material.

Datasets and Metrics. We evaluate on publicly available benchmarks, including the TUM RGB-D [21] and Replica datasets [20]. Replica comprises eight sequences captured in living-room and office scenes populated with common everyday objects. From the TUM RGB-D dataset, we select sequences with both abundant structure and general scenarios to assess performance across diverse indoor environments. For novel-view rendering quality, we use standard photometric metrics, peak signal-to-noise ratio (PSNR), structural similarity index measure (SSIM), and learned perceptual image patch similarity (LPIPS). To quantify camera-pose accuracy, we report the root-mean-square error (RMSE) of the absolute trajectory error (ATE) over all keyframes.

4.2 Novel View Rendering

As listed in Table 1, methods with Gaussian-splatting-based explicit 3D representations (MonoGS, GS-SLAM, LoopSplat, our method) clearly outperform neural fields (NICE-SLAM) in novel view rendering. Furthermore, MonoGS follows closely in PSNR (39.31 dB) but shows slightly higher LPIPS. LoopSplat benefits from explicit loop-closure optimization, maintaining competitive SSIM (0.987) and perceptual quality. Our proposed method achieves the highest average PSNR (39.70 dB) and the best perceptual LPIPS (0.042), indicating superior photometric view synthesis, compared to both systems with and without loop closure modules. Especially, our method performs robustly across all test scenes (*R0–Off3*), showing both high-frequency texture recovery and geometric consistency. The balanced improvement on PSNR, SSIM, and LPIPS confirms that our method not only reconstructs accurate geometry but also preserves fine-grained appearance, yielding photo-realistic novel view synthesis.

Table 1: Rendering comparison on Replica. P: PSNR, S: SSIM, L: LPIPS. The best top-3 results highlighted (1st, 2nd, 3rd).

Seq.	NICE-SLAM			Point-SLAM			Vox-Fusion			SplaTAM			MonoGS			GS-SLAM			LoopSplat			Ours		
	P↑	S↑	L↓	P↑	S↑	L↓	P↑	S↑	L↓	P↑	S↑	L↓	P↑	S↑	L↓	P↑	S↑	L↓	P↑	S↑	L↓	P↑	S↑	L↓
R0	22.12	0.689	0.330	33.38	0.979	0.097	22.39	0.683	0.303	32.60	0.975	0.070	36.44	0.967	0.046	31.56	0.968	0.094	33.07	0.973	0.116	36.67	0.969	0.049
R1	22.47	0.757	0.271	34.10	0.977	0.115	22.36	0.751	0.269	33.55	0.969	0.097	38.55	0.970	0.051	32.86	0.973	0.075	35.32	0.978	0.122	38.67	0.973	0.051
R2	24.52	0.814	0.208	36.32	0.985	0.101	23.92	0.798	0.234	34.83	0.982	0.074	39.52	0.976	0.053	32.59	0.971	0.093	36.16	0.985	0.111	39.63	0.978	0.049
Off0	29.07	0.874	0.229	38.72	0.985	0.089	27.79	0.857	0.241	38.09	0.982	0.088	43.40	0.984	0.031	38.70	0.986	0.050	40.82	0.992	0.085	43.81	0.987	0.029
Off1	30.34	0.886	0.181	39.31	0.987	0.110	29.83	0.876	0.184	39.02	0.982	0.093	43.84	0.984	0.031	41.17	0.993	0.033	40.21	0.990	0.123	44.06	0.987	0.029
Off2	19.66	0.797	0.235	34.22	0.962	0.152	20.33	0.794	0.243	31.95	0.966	0.098	37.32	0.972	0.049	32.36	0.978	0.097	34.67	0.985	0.140	37.74	0.974	0.045
Off3	22.23	0.801	0.209	34.10	0.963	0.119	23.47	0.803	0.213	29.85	0.949	0.119	37.66	0.972	0.042	32.03	0.970	0.110	35.67	0.990	0.096	37.76	0.975	0.041
Off4	24.94	0.856	0.198	34.82	0.981	0.131	25.21	0.847	0.199	31.55	0.951	0.150	37.77	0.968	0.060	32.92	0.968	0.112	37.10	0.989	0.106	39.29	0.975	0.046
Avg	24.42	0.809	0.233	35.62	0.977	0.114	24.41	0.801	0.236	33.89	0.970	0.099	39.31	0.974	0.045	34.27	0.975	0.082	36.63	0.985	0.112	39.70	0.977	0.042

As shown in Fig. 4, we present a qualitative comparison of Gaussian-map fidelity and 3D geometry reconstruction across four sequences. In the *tex_near* scene (Fig. 4a), our method yields a clean and contiguous floor representation with well-structured primitives. In contrast, MonoGS [14] produces numerous floaters. Those floating Gaussians not only degrade visual coherence but also introduce significant noise into the rendered depth maps. A similar issue also appears in the Replica sequences as shown in Fig. 4b. Although high-quality depth maps enhance overall geometric fidelity for both methods, Gaussians located far from the true surface remain misaligned and noisy in the results of MonoGS. More qualitative and quantitative results on the quality of 3D models are provided in the Supplementary Material.

4.3 Camera Pose Estimation

Table 2 reports the camera pose estimation results evaluated using the Absolute Trajectory Error metric on the Replica dataset. Although traditional implicit representation methods such as NICE-SLAM [35] and Vox-Fusion [30] exhibit relatively large trajectory deviations (above 1 cm on average), Gaussian-based SLAM frameworks including MonoGS [14] and GS-SLAM [29] demonstrate improved localization accuracy. Benefiting from the incorporation of loop closure optimization, Loopy-SLAM [12] and LoopSplat [34] demonstrate superior pose accuracy compared to methods without loop closure, within their respective implicit and explicit representation paradigms. Compared to these systems,

Table 2: Comparison of ATE (cm, ↓) on Replica.

Method	R0	R1	R2	Off0	Off1	Off2	Off3	Off4	Avg
NICE-SLAM [35]	0.97	1.31	1.07	0.88	1.00	1.06	1.10	1.13	1.07
Loopy-SLAM [12]	0.24	0.24	0.28	0.26	0.40	0.29	0.22	0.35	0.29
Point-SLAM [18]	0.61	0.41	0.37	0.38	0.48	0.54	0.69	0.72	0.53
Vox-Fusion [30]	1.37	4.70	1.47	8.48	2.04	2.58	1.11	2.94	3.09
SplaTAM [4]	0.31	0.40	0.29	0.47	0.27	0.29	0.32	0.72	0.38
MonoGS [14]	0.46	0.31	0.33	0.47	0.48	0.24	0.18	2.06	0.57
GS-SLAM [29]	0.48	0.53	0.33	0.52	0.41	0.59	0.46	0.70	0.50
LoopSplat [34]	0.28	0.22	0.17	0.22	0.16	0.49	0.20	0.30	0.26
Ours	0.28	0.23	0.26	0.22	0.16	0.13	0.17	0.28	0.22

our proposed Structure Gaussian Splatting SLAM achieves the best overall performance with an average ATE of 0.22 cm, ranking top-1 on five out of eight sequences and within the top-3 on the remaining ones. This result verifies the effectiveness of our structural constraints and optimization strategy in improving camera trajectory consistency and accuracy.

We further evaluate these state-of-the-art GS-SLAM and loop-closure-based methods on the TUM RGB-D benchmark as listed in Table 3. Across all sequences, our method achieves the lowest average ATE and ranks first or second in nearly every individual sequence.

In particular, *fr3/tex_far* and *fr3/tex_near* show that our system remains highly stable even under challenging low-texture and close-range conditions, and competitive results on *fr2/xyz* and *fr2/rpy* demonstrate its robustness to both translational and rotational motions. Table 3 reports the camera trajectory accuracy on the TUM RGB-D dataset in terms of ATE (cm). Our method achieves the lowest average ATE of 1.56 cm, outperforming all competing methods.

Table 3: Comparison of ATE (cm, ↓) on TUM RGB-D.

Method	fr1		fr2		fr3		Avg
	desk	desk2	xyz	rpy	tex_far	tex_near	
Loopy-SLAM [4]	3.79	3.38	1.62	0.40	1.22	2.01	2.07
SplaTAM [4]	3.35	6.54	1.24	0.42	1.43	2.41	2.57
MonoGS [14]	1.48	7.03	1.44	0.57	1.72	1.81	2.34
LoopSplat [34]	2.08	3.54	1.58	1.04	3.71	4.66	2.77
Ours	1.60	4.02	1.47	0.42	0.08	1.76	1.56

4.4 Ablation Study

We evaluate the impact of the key components of our method on both Replica and TUM RGB-D datasets. Additional analyses, including runtime efficiency and alternative presentation schemes of the proposed PGIs, are provided in the Supplementary Material.

Structure Edges of Pose Graph. We evaluate the contributions of the structure closure module in removing the camera pose drift by controlling the frequency of structure closure. As listed in Table 4, the proposed graph exhibits strong robustness. Reducing optimization to once every four keyframes still improves both localization accuracy and rendering quality. Even at once every ten keyframes (50 frames), it outperforms the baseline. At 25-frame intervals, however, limited overlap introduces trajectory drift.

Table 4: Ablation on Structure Closure frequency. We vary the activation interval k and evaluate localization and rendering.

Frequency	Replica R0				Replica Off0			
	ATE↓	PSNR↑	SSIM↑	LPIPS↓	ATE↓	PSNR↑	SSIM↑	LPIPS↓
4	**0.32**	**36.59**	**0.968**	0.051	**0.25**	**43.61**	**0.985**	**0.031**
10	0.37	36.51	0.967	**0.047**	0.32	43.44	0.984	0.032
25	0.53	36.04	0.952	0.054	0.61	42.91	0.978	0.039

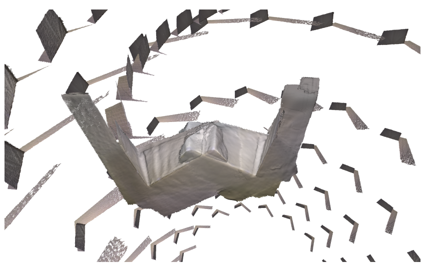
Robustness in Low-textured Scenes. Low-textured environments pose a significant challenge for rendering-based Gaussian Splatting SLAM systems, as photometric gradients vanish, and the camera pose cannot be reliably optimized. Table 5 reports the ATE on the *fr3/notex_far* sequences, where both

Table 5: Camera pose estimation (ATE ↓ [cm]) comparison with state-of-the-art Gaussian Splatting SLAM systems in low-texture sequences.

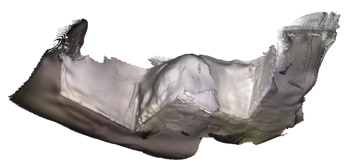
Sequence	MonoGS [14]	LoopSplat [34]	Ours
fr3/notex_far	16.63	8.18	**7.47**
fr3/notex_near	55.59	×	**16.36**

MonoGS [14], our baseline, and LoopSplat [34], a GS-SLAM system with loop closure, exhibits large drift or even fails to produce valid trajectories.

(a) The scene of *fr3/notex_near*



(b) LoopSplat [34]



(c) Ours

Fig. 5: Dense reconstruction in the *fr3/notex_near* sequence of the TUM RGB-D dataset.

In contrast, our method achieves substantially lower ATE, demonstrating strong robustness and stable pose estimation in texture-poor conditions. Fig. 5 shows dense reconstruction results on the *fr3/notex_near* sequence. The scene contains almost no high-frequency texture on the walls and floor, which makes pose estimation and geometry recovery particularly difficult for LoopSplat [34].

Comparison of Plane-ID Representations. We compare the proposed representation with the conventional scalar plane-ID encoding, where each Gaussian stores a single scalar logit to represent its plane identity. In scalar plane-ID representations, numerically adjacent plane IDs (e.g., 16, 17, and 18) introduce artificial ordinal relationships, resulting in ambiguous supervision and unstable optimization, as illustrated in Fig. 6. In contrast, our compact continuous embedding eliminates this undesired ordering effect by representing each plane as a learnable embedding vector, providing more discriminative and stable supervision for plane-aware optimization. Furthermore, the color palette projects the

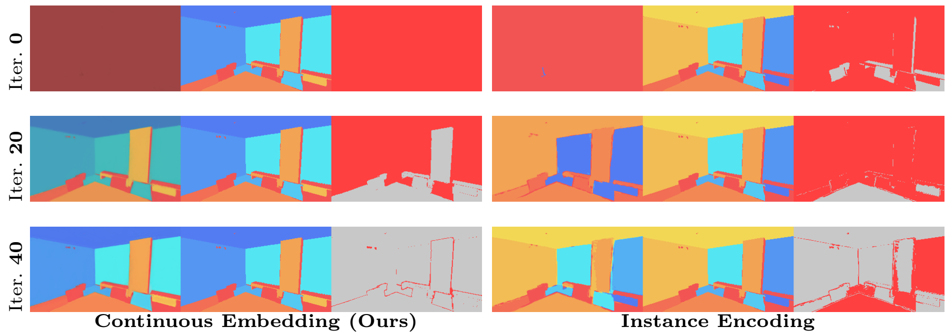


Fig. 6: Planar instance learning on Frame 0 under two different type-encoding strategies: Continuous Embedding and Instance Encoding. We visualize how the rendered planar instance map evolves during the optimization process (Iter. 0 $\rightarrow$ 40). The third column of each method visualizes error map (red: large errors).

learned embeddings into RGB space, enabling dense image-space supervision through differentiable Gaussian rendering.

5 Conclusion

We presented a structure-aware Gaussian SLAM framework that leverages persistent planar structure to improve the stability and accuracy of 3D Gaussian maps. By introducing Planar Gaussian Instances (PGIs), planar surfaces become map entities that support reliable multi-view structural reasoning. Building on this representation, we propose *structure closures* that derive pose constraints from matched non-parallel planes. This enables frequent drift correction even in low-texture or repetitive environments where loop closure is unreliable. We further introduce planar geometry regularization and an adaptive RGB-Depth consistency formulation to stabilize Gaussian optimization and suppress floaters in planar regions, and Sparse Global Viewpoints anchor the sliding-window optimization to maintain global consistency. Experiments on Replica and TUM RGB-D show improvements in both trajectory accuracy and rendering quality over existing GS-SLAM systems. Our work demonstrates that explicitly modeling structural entities provides a promising direction toward geometry-aware Gaussian SLAM.

Acknowledgments

This research / project is supported by the National Research Foundation (NRF) Singapore, under its NRF-Investigatorship Programme (Award ID. NRF-NRFI09-0008), and the Tier 2 grant MOET2EP20124-0015 from the Singapore Ministry of Education.

References

1. Borrmann, D., Elseberg, J., Lingemann, K., Nüchter, A.: The 3d hough transform for plane detection in point clouds: A review and a new accumulator design. 3D Research **2**(2), 1–13 (2011) 4
2. Hu, J., Chen, X., Feng, B., Li, G., Yang, L., Bao, H., Zhang, G., Cui, Z.: Cg-slam: Efficient dense rgb-d slam in a consistent uncertainty-aware 3d gaussian field. In: European Conference on Computer Vision. pp. 93–112. Springer (2024) 4
3. Huang, B., Yu, Z., Chen, A., Geiger, A., Gao, S.: 2d gaussian splatting for geometrically accurate radiance fields. In: ACM SIGGRAPH 2024 Conference Papers. pp. 1–11 (2024) 4
4. Keetha, N., Karhade, J., Jatavallabhula, K.M., Yang, G., Scherer, S., Ramanan, D., Luiten, J.: Splatam: Splat track & map 3d gaussians for dense rgb-d slam. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 21357–21366 (2024) 2, 4, 6, 12, 13
5. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G.: 3d gaussian splatting for real-time radiance field rendering. ACM Trans. Graph. **42**(4), 139–1 (2023) 1, 4
6. Labbe, M., Michaud, F.: Appearance-based loop closure detection for online large-scale and long-term operation. IEEE Transactions on Robotics **29**(3), 734–745 (2013) 4
7. Li, M., Liu, S., Zhou, H., Zhu, G., Cheng, N., Deng, T., Wang, H.: Sgs-slam: Semantic gaussian splatting for neural dense slam. In: European Conference on Computer Vision. pp. 163–179. Springer (2024) 4
8. Li, X., Li, Y., Örnek, E.P., Lin, J., Tombari, F.: Co-planar parametrization for stereo-slam and visual-inertial odometry. IEEE Robotics and Automation Letters **5**(4), 6972–6979 (2020) 2, 4
9. Li, Y., Brasch, N., Wang, Y., Navab, N., Tombari, F.: Structure-slam: Low-drift monocular slam in indoor environments. IEEE Robotics and Automation Letters **5**(4), 6583–6590 (2020) 4
10. Li, Y., Lyu, C., Di, Y., Zhai, G., Lee, G.H., Tombari, F.: Geogaussian: Geometry-aware gaussian splatting for scene rendering. In: European Conference on Computer Vision. pp. 441–457. Springer (2024) 4
11. Li, Y., Yunus, R., Brasch, N., Navab, N., Tombari, F.: Rgb-d slam with structural regularities. In: 2021 IEEE international conference on Robotics and automation (ICRA). pp. 11581–11587. IEEE (2021) 6
12. Liso, L., Sandström, E., Yugay, V., Van Gool, L., Oswald, M.R.: Loopy-slam: Dense neural slam with loop closures. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 20363–20373 (2024) 2, 4, 12
13. Lowe, G.: Sift-the scale invariant feature transform. Int. J **2**(91-110), 2 (2004) 4
14. Matsuki, H., Murai, R., Kelly, P.H., Davison, A.J.: Gaussian splatting slam. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 18039–18048 (2024) 2, 4, 6, 9, 12, 13, 14
15. Mur-Artal, R., Montiel, J.M.M., Tardos, J.D.: Orb-slam: A versatile and accurate monocular slam system. IEEE transactions on robotics **31**(5), 1147–1163 (2015) 3
16. Mur-Artal, R., Tardós, J.D.: Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. IEEE transactions on robotics **33**(5), 1255–1262 (2017) 3, 4
17. Ni, K., Steedly, D., Dellaert, F.: Tectonic sam: Exact, out-of-core, submap-based slam. In: Proceedings 2007 IEEE International Conference on Robotics and Automation. pp. 1678–1685. IEEE (2007) 4

18. Sandström, E., Li, Y., Van Gool, L., Oswald, M.R.: Point-slam: Dense neural point cloud-based slam. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 18433–18444 (2023) 2, 4, 12
19. Schindler, G., Dellaert, F.: Atlanta world: An expectation maximization framework for simultaneous low-level edge grouping and camera calibration in complex man-made environments. In: Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004. vol. 1, pp. I–I. IEEE (2004) 2, 4
20. Straub, J., Whelan, T., Ma, L., Chen, Y., Wijmans, E., Green, S., Engel, J.J., Mur-Artal, R., Ren, C.Y., Verma, S., Clarkson, A., Yan, M., Budge, B., Yan, Y., Pan, X., Yon, J., Zou, Y., Leon, K., Carter, N., Briales, J., Gillingham, T., Mueggler, E., Pesqueira, L., Savva, M., Batra, D., Strasdat, H.M., Nardi, R.D., Goesele, M., Lovegrove, S., Newcombe, R.A.: The replica dataset: A digital replica of indoor spaces. arXiv preprint arXiv: 1906.05797v1 (2019) 11
21. Sturm, J., Engelhard, N., Endres, F., Burgard, W., Cremers, D.: A benchmark for the evaluation of rgb-d slam systems. In: 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems. pp. 573–580 (2012) 11
22. Tateno, K., Tombari, F., Laina, I., Navab, N.: Cnn-slam: Real-time dense monocular slam with learned depth prediction. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 6243–6252 (2017) 4
23. Teed, Z., Deng, J.: Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras. Advances in neural information processing systems **34**, 16558–16569 (2021) 4
24. Tran, A.T., Kosecka, J.: Varsplat: Uncertainty-aware 3d gaussian splatting for robust rgb-d slam. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 26072–26082 (2026) 4
25. Von Gioi, R.G., Jakubowicz, J., Morel, J.M., Randall, G.: Lsd: A line segment detector. Image Processing On Line **2**, 35–55 (2012) 4
26. Wang, C., Guo, X.: Plane-based optimization of geometry and texture for rgb-d reconstruction of indoor scenes. In: 2018 International Conference on 3D Vision (3DV). pp. 533–541. IEEE (2018) 6
27. Whelan, T., Leutenegger, S., Salas-Moreno, R.F., Glocker, B., Davison, A.J.: Elasticfusion: Dense slam without a pose graph. In: Robotics: science and systems. vol. 11, p. 3. Rome, Italy (2015) 3
28. Wu, J., Sha, Y., Jiang, B., Tan, J.: Dsine: deep structural influence learning via network embedding. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 33, pp. 10065–10066 (2019) 4
29. Yan, C., Qu, D., Xu, D., Zhao, B., Wang, Z., Wang, D., Li, X.: Gs-slam: Dense visual slam with 3d gaussian splatting. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 19595–19604 (2024) 12
30. Yang, X., Li, H., Zhai, H., Ming, Y., Liu, Y., Zhang, G.: Vox-fusion: Dense tracking and mapping with voxel-based neural implicit representation. In: 2022 IEEE International Symposium on Mixed and Augmented Reality (ISMAR). pp. 499–507. IEEE (2022) 12
31. Yugay, V., Li, Y., Gevers, T., Oswald, M.R.: Gaussian-slam: Photo-realistic dense slam with gaussian splatting. arXiv preprint arXiv:2312.10070 (2023) 2, 4
32. Yunus, R., Li, Y., Tombari, F.: Manhattanslam: Robust planar tracking and mapping leveraging mixture of manhattan frames. In: 2021 IEEE International Conference on Robotics and Automation (ICRA). pp. 6687–6693. IEEE (2021) 2, 4

33. Zhao, L., Huang, S., Dissanayake, G.: Linear slam: A linear solution to the feature-based and pose graph slam based on submap joining. In: 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems. pp. 24–30. IEEE (2013) 4
34. Zhu, L., Li, Y., Sandström, E., Huang, S., Schindler, K., Armeni, I.: Loopsplat: Loop closure by registering 3d gaussian splats. arXiv preprint arXiv:2408.10154 (2024) 2, 4, 12, 13, 14
35. Zhu, Z., Peng, S., Larsson, V., Xu, W., Bao, H., Cui, Z., Oswald, M.R., Pollefeys, M.: Nice-slam: Neural implicit scalable encoding for slam. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 12786–12796 (2022) 12
36. Zhu, Z., Fang, Y., Li, X., Yan, C., Xu, F., Yuen, C., Li, Y.: Robust gaussian splatting slam by leveraging loop closure. arXiv preprint arXiv:2409.20111 (2024) 2