SING3R-SLAM: Submap-based Indoor Monocular Gaussian SLAM with 3D Reconstruction Priors

Kunyi Li[1,3] Michael Niemeyer[2] Sen Wang[1,3] Stefano Gasperini[1,3,4]
Nassir Navab[1,3] Federico Tombari[1,2]

[1]Technical University of Munich [2]Google [3]Munich Center for Machine Learning [4]VisualAIs

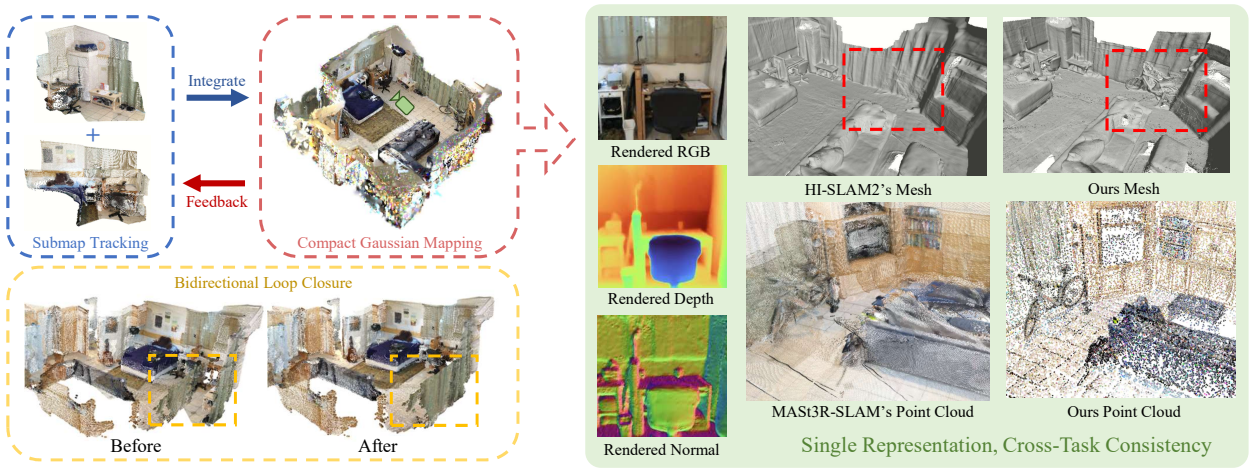

Figure 1. **SING3R-SLAM** is a submap-based monocular SLAM system enhanced by 3D priors. Left: our key modules, where tracking produces locally accurate point maps, mapping fuses them into a compact global representation, and joint optimization further refines poses and geometry, aided by bidirectional loop closure. Right: the resulting Gaussian map supports multiple downstream tasks with global geometry consistency, extending our method beyond pose estimation.

Abstract

Recent advances in dense 3D reconstruction enable the accurate capture of local geometry; however, integrating them into SLAM is challenging due to drift and redundant point maps, which limit efficiency and downstream tasks, such as novel view synthesis. To address these issues, we propose SING3R-SLAM, a globally consistent and compact Gaussian-based dense RGB SLAM framework. The key idea is to combine locally consistent 3D reconstructions with a unified global Gaussian representation that jointly refines scene geometry and camera poses, enabling efficient and versatile 3D mapping for multiple downstream applications. SING3R-SLAM first builds locally consistent submaps through our lightweight tracking and reconstruction module, and then progressively aligns and fuses them into a global Gaussian map that enforces cross-view geometric consistency. This global map, in turn, provides feedback to correct local drift and enhance the robustness of tracking. Extensive experiments demonstrate that SING3R-SLAM achieves state-of-the-art tracking, 3D reconstruction, and novel view rendering, resulting in over 12% improvement in tracking and producing finer, more detailed geometry, all while maintaining a compact and memory-efficient global representation on real-world datasets.

1. Introduction

Visual simultaneous localization and mapping (vSLAM) lies at the core of modern robotics [22, 31, 32] and augmented reality [2, 29] applications, enabling systems to perceive and understand their surroundings [1, 3, 12, 17, 19]. With advances in hardware and algorithmic design, robust and accurate visual SLAM [14, 16, 21, 22, 31] has become increasingly feasible. Yet, developing a plug-and-play dense SLAM system that operates reliably in the wild remains an open challenge with the potential to significantly advance spatial perception [12, 13, 19, 24, 34–36] and scene

understanding [1, 17, 25, 26]. Achieving both precise camera pose estimation and globally consistent dense reconstruction solely with a minimal monocular setup remains difficult, despite numerous handcrafted [9, 27] and data-driven [13, 31] priors that have been explored in this domain. Previous methods that use 2D priors, such as optical flow [8, 31] or feature matching [13, 22], achieve remarkable tracking performance. Yet, disentangling pose estimation and global geometry remains challenging, as lighting conditions or object appearances may vary across different views. Single-view 3D priors, such as monocular depth [37] and surface normals [6], can provide local geometry estimation, though they suffer from scale ambiguity and lack cross-view consistency.

Recent advances in large-scale 3D pretraining have led to powerful two-view reconstruction models such as DUSt3R [36] and MASt3R [13], which establish strong structure-from-motion priors. Multi-view extensions like Spann3R [33], CUT3R [35], and VGGT [34] enhance local geometric consistency, but still lack global coherence. Sequential variants, including SLAM3R [18], MASt3R-SLAM [23], and VGGT-SLAM [20], further incorporate temporal alignment, but their reliance on pairwise matching or manifold optimization limits scalability and global consistency. Overall, while these SLAM frameworks achieve accurate local geometry and improved poses, they remain confined to locally consistent submaps without optimizing the underlying point maps. Consequently, misalignment persists across views, and global consistency degrades over long sequences. Importantly, the lack of a unified global model hinders downstream tasks such as novel view synthesis (NVS), which has become an increasingly popular metric for evaluating global 3D reconstruction fidelity and multi-view consistency.

3D Gaussian-based SLAM methods [10, 16, 21, 42] provide a globally consistent scene representation using 3D Gaussian Splatting (3DGS) [11], enabling joint optimization of camera poses and geometry for consistent reconstruction and high-quality novel view synthesis via differentiable rendering. However, most existing systems rely on RGB-D input, which limits their applicability. RGB-only variants such as Splat-SLAM [28] and HI-SLAM2 [42] rely on pretrained depth and normal estimators, with the resulting depth often lacking accurate scale and global consistency. Moreover, these pipelines rely on external tracking modules for camera poses, thereby disentangling scene geometry reconstruction from pose estimation. Additionally, integrating multiple pretrained models also increases computational costs.

To address these challenges, we propose SING3R-SLAM (see Fig. 1), a submap-based RGB SLAM framework that unifies locally accurate 3D priors with a globally consistent Gaussian representation. The key idea is to let local reconstruction and global mapping reinforce each other: the Sub-Track3R module produces geometrically locally-reliable submaps and efficiently aligns them through overlapping frames, providing a strong initialization for the Gaussian Mapper. The mapper, in turn, refines camera poses and corrects geometric drift by jointly optimizing the global Gaussian map, which is continuously fed back to stabilize tracking over long sequences. Our method simultaneously supports accurate tracking, globally consistent reconstruction, and high-quality novel view synthesis. In summary, our main contributions are as follows:

- A submap-based dense RGB SLAM framework tightly integrating Sub-Track3R's local geometry with a globally consistent, compact mapper for accurate pose estimation and reconstruction.
- An inter- and intra-submap registration approach to correct pose and geometry errors, enhancing tracking and reconstruction performance.
- A novel bidirectional loop closure module, where a pointmap-based loop correction refines large trajectory drift, and the Gaussian map further refines camera poses and scene geometry.

2. Related Work

2.1. 3D Reconstruction and SLAM

Recent advances [13, 33–36] in 3D reconstruction priors have significantly improved monocular geometry estimation and inspired new directions for visual SLAM. DUSt3R [36] and its successor MASt3R [13] pioneer two-view 3D reconstruction, achieving impressive geometric accuracy by leveraging large-scale 3D datasets and correspondence-based priors. However, as they operate on only two input views, these methods require additional global alignment [5] to handle long sequences and maintain geometric consistency. To overcome this limitation, Spann3R [33] introduces a memory bank to extend reconstruction from two to multiple views. Despite improved temporal consistency, it remains constrained to low-resolution inputs (224×224) and only performs well for up to five frames. Moreover, the reconstructed point maps often exhibit slight scale inconsistencies between frames. CUT3R [35] and VGGT [34] further extend multi-view reconstruction and achieve better geometric stability. Both methods still experience severe drift when applied to longer sequences, and VGGT does not process inputs sequentially, limiting its applicability to continuous SLAM pipelines.

Building on these priors, SLAM3R [18] is the first system to integrate DUSt3R [36] into a sequential framework, performing pairwise two-view matching for pose estimation. However, its overall performance remains limited. Similarly, MASt3R-SLAM [23] employs feature matching between adjacent frames for continuous two-view align-

ment. While this enables sequential tracking, its reliance on frame-by-frame matching makes it inefficient and susceptible to drift over long trajectories. VGGT-SLAM [20] further leverages VGGT [34] to reconstruct submaps from multiple frames and optimize their poses on the $SL(4)$ manifold. Although this formulation improves global alignment, it primarily focuses on pose optimization instead of global geometry reconstruction. ViSTA-SLAM [40] introduces Symmetric Two-view Association (STA) with shared weights in tracking, but like previous methods, it leaves the reconstructed point maps as unrefined outputs from the reconstruction network.

Overall, existing 3D reconstruction-based SLAM systems primarily emphasize pose estimation while leaving the reconstructed point maps unrefined. As a result, local geometric errors persist and global consistency is not enforced, causing drift in long sequences [18] and reducing reconstruction quality in large-scale scenes. In addition, storing dense per-frame point maps is memory-intensive and limits scalability [20, 23]. In contrast, our proposed SING3R-SLAM addresses these by integrating locally accurate submaps with a globally consistent Gaussian map, which refines both poses and point maps, reduces drift, and provides a compact, memory-efficient representation suitable for long-sequence tracking and downstream tasks such as novel view synthesis.

2.2. 3D Gaussians-based SLAM

Recent 3D Gaussians-based SLAM methods leverage the expressive power of 3D Gaussian Splatting [11] to represent the scene as a continuous and globally consistent map. Unlike point representations [34, 36], the Gaussian map provides a compact global model. This formulation allows differentiable rendering and efficient global optimization, making it highly suitable for dense SLAM.

Since the training of Gaussian models typically requires registered point clouds as input, most existing Gaussian SLAM systems [10, 21, 38] are designed for RGB-D settings. SplaTAM [10] and Gaussian Splatting SLAM [21] are representative examples that utilize depth input to initialize and update Gaussian parameters in real time, achieving accurate pose tracking and high-quality reconstruction but limiting applicability to depth-equipped sensors.

More recent methods [7, 15, 28, 42, 43] extend Gaussian SLAM to RGB-only input. Splat-SLAM [28] and HI-SLAM2 [42] employ off-the-shelf depth and normal estimators to infer per-view geometry and use external tracking modules (e.g., DROID-SLAM [31]) for camera pose estimation. Although this design enables monocular operation, the predicted depth lacks scale information and global consistency, leading to cumulative drift over time. Moreover, relying on multiple pretrained modules results in fragmented pipelines with increased computational overhead.

While Gaussian-based SLAM offers a compact and globally consistent scene representation, existing methods often depend on depth sensors [10, 21, 38] or multiple external modules to separately estimate geometry and camera poses [28, 42]. This separation limits joint optimization and can lead to scale ambiguities, cumulative drift, and fragmented pipelines. However, our SING3R-SLAM combines locally accurate 3D reconstruction priors with a global Gaussian map, enabling joint refinement of poses and global scene geometry.

3. Method

3.1. Local Sub-Track3R

SLAM systems take sequential RGB images $\{C_l\}_{l=0}^{N}$ as input and estimate camera poses $\{T_l\}_{l=0}^{N} \in SE(3)$ and a reconstruction of the scene. MASt3R-SLAM [23] generates per-frame point maps through pairwise image matching, which is inefficient and computationally expensive. VGGT-SLAM [20] instead reconstructs submaps from frame batches and aligns them via optimization on the $SL(4)$ manifold, but its non-sequential batch processing and complex alignment limit scalability. In contrast, our method employs a sequential 3D encoder to build submaps and introduces a simple yet efficient inter-submap registration for continuous tracking.

Local Submap Reconstruction. As shown in the top-middle of Fig. 2, we denote the sequential input images as $\{C_{i,j}\}_{j=0}^{K} \in G_i$, where G_i represents the i-th submap, $i = 0, \ldots, \lfloor N/K \rfloor$, and K is the number of frames per submap. To ensure temporal continuity, we introduce an overlapping frame between consecutive submaps such that the last frame $C_{i-1,K}$ of G_{i-1} is identical to the first frame $C_{i,0}$ of G_i. Each frame is then processed by an off-the-shelf sequential 3D encoder to predict dense point maps and corresponding camera poses, which together constitute the submap representation:

$$\{X_{i,j}^{self}\}_{j=0}^{K}, \{T_{i,j}^{i-th}\}_{j=0}^{K} = Encoder(\{C_{i,j}\}_{j=0}^{K}) \quad (1)$$

where $\{X_{i,j}^{self}\}_{j=0}^{K} \in G_i$ is the point map in self coordinates, and $\{T_{i,j}^{i-th}\}_{j=0}^{K} \in G_i$ represents the corresponding camera poses in the i-th local coordinates. Depth maps can be extracted from local point maps as $\{D_{i,j}\}_{j=0}^{K}$.

Inter-Submap Registration. Point maps and poses from the 3D encoder are all in local coordinates. While VGGT-SLAM [20] uses an elaborate optimization to align the point maps and poses into world coordinates, we propose a simple and optimization-free method. Assuming the previous submap G_{i-1} is already in world coordinate, we transfer $\{X_{i,j}^{self}\}_{j=0}^{K}$ and $\{T_{i,j}^{i-th}\}_{j=0}^{K}$ by:

$$\begin{aligned} T_{i,j}^{world} &= T_{i-1,K}^{world} \left(T_{i,0}^{i-th}\right)^{-1} T_{i,j}^{i-th}, \\ X_{i,j}^{world} &= T_{i,j}^{world}\, s_i X_{i,j}^{self}, \end{aligned} \quad (2)$$

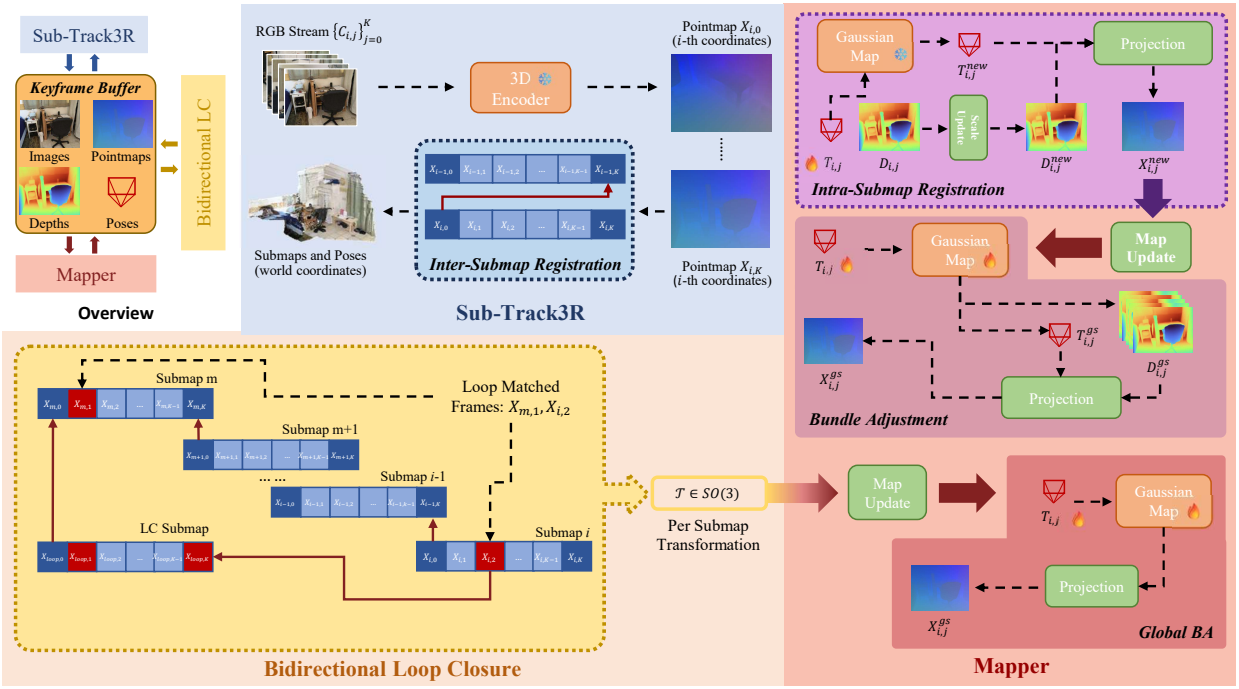

Figure 2. **Overview.** Our system comprises three main components: Sub-Track3R (top-middle), Mapper (right), and Loop Closure (bottom-left). The top-left shows that these components interact and exchange data through the keyframe buffer to maintain consistency. The Sub-Track3R performs tracking between submaps, predicting point maps and local poses that are aligned into the world coordinate system via inter-submap registration. The Mapper employs a Gaussian model as a globally consistent and compact scene representation, jointly optimizing Gaussians and poses to achieve coherent geometry and improved reconstruction quality. In the Loop Closure, point map-based correction reduces large trajectory drift, and the updated Gaussian map further refines poses for globally consistent reconstruction.

where $X_{i,j}^{world} \in G_i$ denotes the world-coordinate point map of frame j in the current submap, and $T_{i-1,K}^{world} \in G_{i-1}$ is the world-coordinate pose of the last frame in the previous submap. Since the 3D encoder may generate point maps with varying scales for different input batches, we introduce a scale correction factor $s_i = \exp\Big(\log(D_{i-1,K}) - \log(D_{i,0})\Big)$, which compensates for scale inconsistencies between adjacent submaps.

While the Sub-Track3R provides inter-submap accurate poses and fine-grained local geometry, intra-submap misalignment remains and accumulates over long sequences. A global representation is therefore needed to jointly refine per-frame geometry and camera poses.

3.2. Global Mapper

Previous methods [20] reduce intra-submap pose errors only after loop closure, and their dense per-frame point maps often suffer from scale inaccuracies, inter-frame inconsistencies, as shown in Fig. 4, and high memory costs. To address this, we use a lightweight Gaussian map initialized from Sub-Track3R point maps, which jointly refines camera poses and scene geometry via multi-view optimization. Unlike in HI-SLAM2 [42], where mapping is independent of tracking, we feed the updated poses and geometry back to the Sub-Track3R, forming a tightly coupled SLAM framework with improved accuracy and consistency.

Global Map Representation. As shown in the Fig. 2, we employ a global Gaussian map as a compact and continuous scene representation, as it models the scene with fewer points than dense point maps while preserving geometry. Being defined in a global coordinate system and optimized via multi-view rendering naturally enforces global cross-view consistency. Each 3D Gaussian $\mathcal{G}_k = \{\mu_{i,k}, \Sigma_{i,k}, \alpha_{i,k}, \mathbf{c}_{i,k}\}$ consists of a mean position $\mu_{i,k} \in \mathbb{R}^3$, a covariance matrix $\Sigma_{i,k} = R_{i,k}S_{i,k}S_{i,k}^{\top}R_{i,k}^{\top}$, $R_{i,k} \in SO(3), S_{i,k} = [s_k^0, s_k^1, s_k^2] \in \mathbb{R}^3$, an opacity $\alpha_{i,k}$ and a color $\mathbf{c}_{i,k}$. The first subscript i indicates the i-th submap from which this Gaussian was generated. The Gaussian map $\mathcal{M} = \{\mathcal{G}_k\}_{k=1}^{N_g}$ serves as a global, differentiable scene representation. Given a camera pose $T_{i,j}^{world}$, through a standard Gaussian Splatting [11, 39] process, we can render the corresponding color map $\hat{C}_{i,j}$, depth map $\hat{D}_{i,j}$, normal map $\hat{N}_{i,j}$ and silhouette map $\mathcal{A}_{i,j}$. The Gaussian model is initialized in the world coordinates defined by the pose of the first frame $T_{0,0}^{world}$, using the corresponding point map $X_{0,0}^{world}$.

Intra-Submap Registration. For the following submaps

$\{X_{i,j}^{world}\}$ and their corresponding poses $\{T_{i,j}^{world}\}$, instead of directly using them to initialize the new Gaussians, we first perform the intra-submap registration to correct the pose and scale errors within the submap. Given an incoming view $\hat{\mathcal{V}}_{ij} = \{C_{i,j}, D_{i,j}, X_{i,j}^{world}, T_{i,j}^{world}\}$, we minimize the photometric and geometric losses:

$$\min_{T_{i,j}^{world}} \mathcal{L}_C + \lambda_{scaleD}\mathcal{L}_{scaleD}, \tag{3}$$

where $\mathcal{L}_C = \|\mathcal{A}_{i,j}C_{i,j} - \mathcal{A}_{i,j}\hat{C}_{i,j}\|_1$ and $\mathcal{A}_{i,j}$ is used to ignore empty pixels where there's no Gaussians during rendering [10]. We apply the scale-invariant depth loss to account for the potential scale inaccuracies in $D_{i,j}$:

$$\mathcal{L}_{scaleD} = \sum \left(\log \mathcal{A}_{i,j}\hat{D}_{i,j} - \log \mathcal{A}_{i,j}D_{i,j}\right)^2 - \left(\sum \left(\log \mathcal{A}_{i,j}\hat{D}_{i,j} - \log \mathcal{A}_{i,j}D_{i,j}\right)\right)^2. \tag{4}$$

After pose refinement, we update the point map with the new camera pose $T_{i,j}^{new}$ and scaled depth:

$$\begin{aligned} D_{i,j}^{new} &= \exp\left(\log(\mathcal{A}_{i,j}\hat{D}_{i,j}) - \log(\mathcal{A}_{i,j}D_{i,j})\right)D_{i,j}, \\ X_{i,j}^{new} &= T_{i,j}^{new}\,\pi^{-1}(D_{i,j}^{new}, P), \end{aligned} \tag{5}$$

where P is the camera intrinsic and π represents the projection operator. The new training view is registered as $\mathcal{V}_{ij} = \{C_{i,j}, T_{i,j}^{new}, D_{i,j}^{new}, X_{i,j}^{new}, \mathcal{A}_{i,j}\}$. We use the superscript *new* to indicate the updated values.

Map Update and Bundle Adjustment. With the updated point map $X_{i,j}^{new}$ and silhouette masks $\{\mathcal{A}_{i,j}\}_{j=0}^{K}$, we augment the Gaussian map by adding new Gaussians exclusively in the previously uncovered regions. We apply bundle adjustment to jointly update the Gaussian map and poses:

$$\min_{\mathcal{M},T} \sum_{\{\mathcal{V}_m\}_{m=ij-W}^{ij}} \mathcal{L}_{pho} + \lambda_D\mathcal{L}_D + \lambda_{DN}\mathcal{L}_{DN} + \lambda_S\mathcal{L}_S. \tag{6}$$

We optimize through multi-view rendering, ensuring global consistency to avoid overfitting to individual views, and W is the size of the rendering window. For a view $\mathcal{V}_{i,j}$, $\mathcal{L}_{pho} = \|C_{i,j} - \hat{C}_{i,j}\|_1 + SSIM(C_{i,j}, \hat{C}_{i,j})$ is the photometric loss, $\mathcal{L}_D = \|1/D_{i,j}^{new} - 1/\hat{D}_{i,j}\|_1$ is the inverted depth loss. $\mathcal{L}_{DN} = (1 - N_{i,j} \cdot \bar{N}_{i,j})$ is the depth-normal loss, where the pseudo normal map $N_{i,j}$ is converted from $D_{i,j}^{new}$, and the depth-normal map $\bar{N}_{i,j}$ is converted from $\hat{D}_{i,j}$. Finally, the scale loss $\mathcal{L}_S = \sum_{j=0}^{2}(s_k^j - \bar{s}_k)$ is used to prevent artifacts due to excessively slender Gaussians, and $\bar{s}_i$ is the mean scale value.

Submap Update. After optimizing the global map, we obtain the updated camera poses $T_{i,j}^{gs}$ and rendered depth $\hat{D}_{i,j}$. We then update the point maps and camera poses for the next tracking iteration of the Sub-Track3R as follows:

$$\begin{aligned} D_{i,j}^{gs} &= \exp\left(\log(\hat{D}_{i,j}) - \log(D_{i,j})\right)D_{i,j}, \\ X_{i,j}^{gs} &= T_{i,j}^{gs}\,\pi^{-1}(D_{i,j}^{gs}, P). \end{aligned} \tag{7}$$

Here, $D_{i,j}^{gs}, X_{i,j}^{gs}$, and $T_{i,j}^{gs}$ will replace the previous depth $D_{i,j}$, point map $X_{i,j}^{world}$ and pose $T_{i,j}^{world}$ in the Sub-Track3R for the next tracking iteration. These updated poses and point maps are now expressed in the globally consistent geometry and also help mitigate accumulated errors within each submap.

3.3. Backend Optimization

Bidirectional Loop Closure. We further introduce a loop closure mechanism to suppress long-term drift. Unlike previous methods [20, 23, 42] which rely on either feature correspondence or bag-of-words, our approach leverages the globally consistent Gaussian map as a reference to perform loop closure. For each frame, we obtain high-quality point maps through Sub-Track3R and the Gaussian mapper, which allows us to construct the covisibility graph using reprojection without requiring explicit feature matching and detect a loop simply based on the overlapping ratio. Details of the loop detection are provided in the Appendix.

As illustrated at the bottom of Fig. 2, suppose the current frame $C_{i,j}$ is detected to form a loop with frame $C_{m,n}$ from a previous submap G_m. We construct a new submap $\{C_{\text{loop},j}\}_{j=0}^{K} \in G_{\text{loop}}$, where $\{C_{\text{loop},j}\}_{j=0}^{K-1} = \{C_{m,n}\}_{n=0}^{K-1}$ and $C_{\text{loop},K} = C_{i,j}$. Using the 3D encoder, we generate point maps and corresponding poses. Then, we transform them into world coordinates via the overlapping frame $C_{m,n}$ using Eq. 2, yielding $\{X_{\text{loop},j}^{\text{world}}\}_{j=0}^{K} \in G_{\text{loop}}$ and the corresponding poses $\{T_{\text{loop},j}^{\text{world}}\}_{j=0}^{K} \in G_{\text{loop}}$. To close the loop, we then minimize the following Euclidean distance:

$$\min_{\mathcal{T}} \sum_{t=0} \Big(\left\|\mathcal{T}_{t-1}(X_{t-1,K}) - \mathcal{T}_t(X_{t,0})\right\|_2^2 + \left\|\mathcal{T}_i(X_{i,j}^{\text{world}}) - \mathcal{T}_m(X_{\text{loop},K}^{\text{world}})\right\|_2^2 \Big), \tag{8}$$

where $\mathcal{T} \in SE(3)$ denotes the rigid transformations applied to each submap. The first term enforces consistency between adjacent submaps, while the second ensures the closure of the detected frame.

While loop closure based on point maps can partially correct drift between submaps, it does not achieve highly precise alignment. By leveraging the Gaussian map as a globally consistent scene representation, we perform joint optimization of both the map and camera poses with greater accuracy, refining local and global geometric consistency across the entire sequence. Once the submap transformations are optimized, we apply them to the Gaussians to update the global map.

$$\mu'_{i,k} = \mathcal{T}_i(\mu_{i,k}),\ R'_{i,k} = \mathcal{T}_i(R_{i,k}), \tag{9}$$

where i represents the corresponding i-th submap.

We then perform a global pose adjustment using Eq. 3 across all views to jointly refine the camera poses with respect to the updated Gaussian map, effectively reducing accumulated drift and enforcing global geometric consistency.

Method	**Avg.**	chess	fire	heads	office	pumpkin	redkitchen	stairs
CUT3R [35]	47.7	74.3	22.6	36.3	66.4	54.6	38.1	41.3
MASt3R-SLAM [23]	6.6	6.3	4.6	2.9	10.3	11.2	7.4	3.2
VGGT-SLAM [20]	6.7	3.6	2.8	1.8	10.3	13.3	5.8	9.3
ViSTA-SLAM [40]	5.5	7.3	3.5	2.8	5.5	12.9	3.5	2.9
HI-SLAM2 [42]	5.5	3.8	3.1	2.6	8.5	14.2	4.0	2.4
SING3R-SLAM (Ours)	4.8	3.4	2.7	3.5	7.1	8.7	3.5	4.4

Table 1. **Quantitative Comparison of Camera Tracking Accuracy on 7-scenes [30].** Our method achieves the best tracking performance among both 3D reconstruction-based and Gaussian-based approaches, reducing the average ATE by 12%.

After optimization, the refined poses and depth maps are reprojected to generate updated point maps following Eq. 7, which are subsequently fed back to the Sub-Track3R module for the next tracking and loop closure iteration.

Global Bundle Adjustment. We perform a global bundle adjustment to jointly refine all camera poses and the Gaussian map together with tracking and mapping by:

$$\min_{\mathcal{M},T} \sum_{\{\mathcal{V}_m\}_{m=0}^{N}} \mathcal{L}_{pho} + \lambda_D \mathcal{L}_D + \lambda_{DN} \mathcal{L}_{DN} + \lambda_N \mathcal{L}_N. \quad (10)$$

where the normal loss is $\mathcal{L}_N = (1 - N_{i,j} \cdot \hat{N}_{i,j})$ and $\hat{N}_{i,j}$ is the rendered normal map. This prevents the Gaussian model from overfitting to the currently observed views while further reducing accumulated drift and enforcing a globally consistent geometry. Using Eq. 7, the updated poses and depth maps are then reprojected to generate refined point maps for the Sub-Track3R. Note that this optimization can run in parallel with the previous modules.

4. Experiments

4.1. Experimental Settings

Datasets. We comprehensively evaluate our method on two public datasets: 7-scenes [30], ScanNet-v2 [4].

Baselines. We compare our proposal with several state-of-the-art methods. Among 3D reconstruction-based SLAM methods, we compare against MASt3R-SLAM [23], VGGT-SLAM [20] and ViSTA-SLAM [40]. For pose estimation, we additionally include CUT3R [35]. As for Gaussian-based SLAM methods, we compare against MonoGS [21], Splat-SLAM [28], and HI-SLAM2 [42]. Finally, we compare with two RGB-D methods, SplaTAM [10] and Gaussian-SLAM [38], in terms of rendering quality.

Metrics. To evaluate our method, we follow standard practice and report surface accuracy using Accuracy, Completeness, and Chamfer Distance. Visual fidelity of synthesized novel views is measured with PSNR, SSIM, and LPIPS [41], while Absolute Trajectory Error (ATE) is used for pose evaluation.

Implementation. CUT3R [35] is used as our 3D encoder and RaDe-GS [39] is used as our Gaussian rasterizer. We train on a single NVIDIA RTX 4090 GPU. For tracking, we choose $K = 6$ for each submap. For mapping, we set $\lambda_{scaleD} = 10, \lambda_{DN} = 0.05, \lambda_N = 0.05, \lambda_S = 10$ and $\lambda_D = 5$ in the map update and $\lambda_D = 0.5$ in the global bundle adjustment. Refer to the Appendix for more details.

Method	PSNR ↑	SSIM ↑	LPIPS ↓
RGB-D input			
SplaTAM [10]	20.42	0.78	0.38
Gaussian-SLAM [38]	27.67	0.92	0.25
RGB input			
Splat-SLAM [28]	29.48	0.85	0.18
HI-SLAM2 [42]	29.27	0.88	0.24
SING3R-SLAM (Ours)	30.47	0.89	0.21

Table 2. **Quantitative Comparison of Average Rendering Performance on ScanNet-v2 [4].**

4.2. Pose Estimation

We evaluate the camera tracking accuracy of our system against both 3D reconstruction-based and Gaussian-based approaches. As reported in Table 1, SING3R-SLAM achieves the best average tracking performance on the 7-scenes dataset, demonstrating a substantial improvement of over 12% in average ATE compared to existing methods. The performance gain is particularly pronounced on challenging scenes such as *pumpkin*, where our approach significantly outperforms all baselines. These results highlight the effectiveness of integrating locally consistent submaps with a globally optimized Gaussian map, which not only reduces drift but also ensures robust tracking across diverse and complex indoor environments.

4.3. Rendering

Novel view synthesis has become an important metric for evaluating 3D reconstruction, reflecting both geometric accuracy and multi-view consistency. In Table 2, compared to RGB-D and RGB-only Gaussian SLAM methods, our approach achieves the best performance by leveraging a compact, globally consistent Gaussian map. This enables high-fidelity rendering while preserving geometric consistency.

4.4. Geometry Reconstruction

We evaluate 3D reconstruction quality on the 7-Scenes dataset [30] (Table 3). Our method achieves competi-

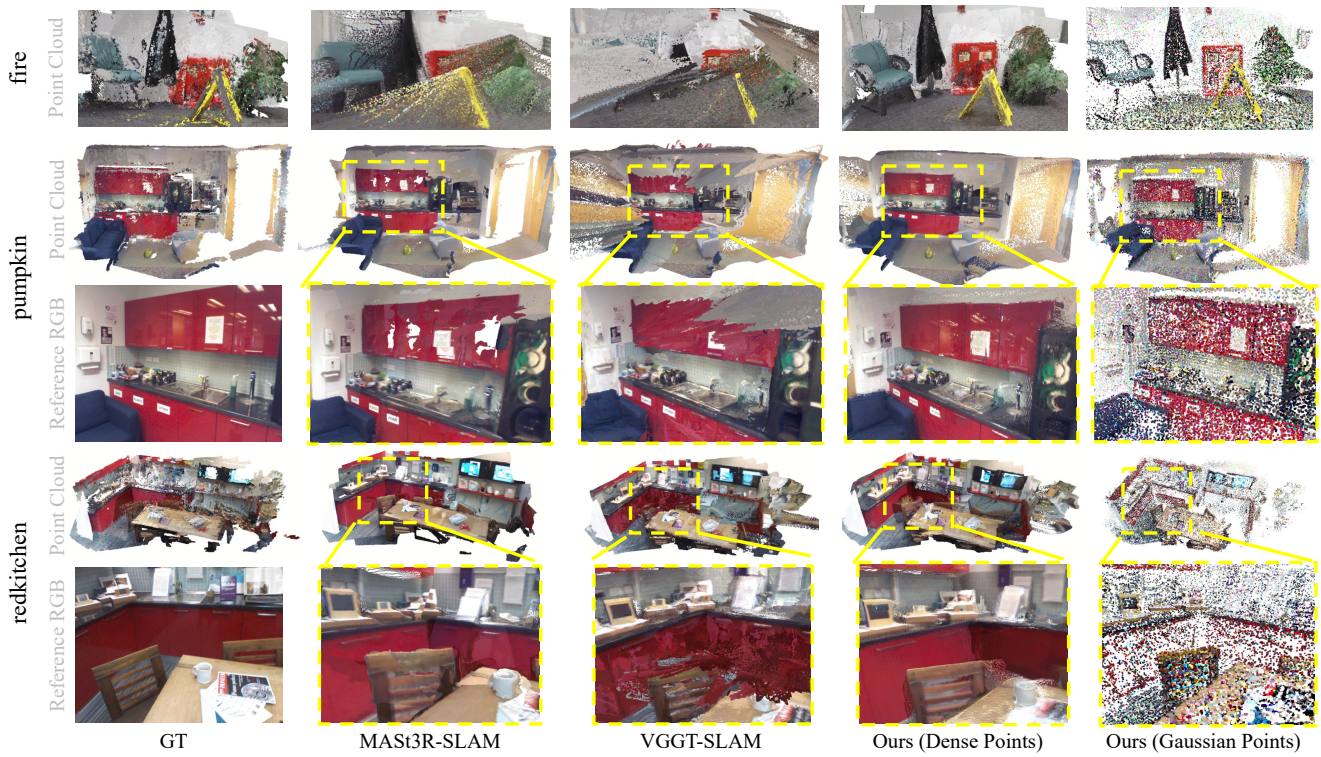

Figure 3. **Qualitative Comparison of Reconstructed Point Clouds on 7-scenes [30].** We show the reconstructed point clouds with zoomed-in views for all methods. Our approach provides a compact Gaussian representation that is much cleaner and captures object geometry in detail, as illustrated in the last column. In contrast, other 3D reconstruction-based methods often produce many redundant points, which degrade visual quality. Moreover, our dense point reconstruction preserves geometric structures more accurately.

Figure 4. **Qualitative Comparison of Reconstructed Point Clouds on *office*.** Left: RGB images from different views. Middle: VGGT-SLAM. Right: SING3R-SLAM (Ours). Our approach accurately aligns the wall and table across views, whereas VGGT-SLAM produces misaligned and overlapping geometry.

tive results, though the numerical metrics should be interpreted cautiously because the ground-truth point clouds are incomplete (Fig. 3), which penalizes our more complete reconstructions. Qualitatively, our method consistently produces accurate and complete geometry. In Fig. 3 (*pumpkin*), MASt3R-SLAM [23] misses the reflective cabinet doors and VGGT-SLAM [20] introduces structural errors, while our reconstruction remains faithful. Compared with HI-SLAM2 [42], our approach also better preserves fine details, such as the bicycles in ScanNet-v2 scene_*0000* (Fig. 5). As shown in Fig. 4, prior methods suffer from wall misalignment due to the lack of a global map, whereas ours maintains globally consistent and locally precise geometry. These results highlight the benefit of combining locally consistent submaps with a globally optimized Gaussian map for high-quality reconstruction.

4.5. Ablations and Performance Analysis

All ablations and performance analysis are conducted on the scene_*0059* of the ScanNet-v2 [4] dataset.

Performance Analisys. We evaluate the contributions of key components in SING3R-SLAM in Table 4, where (g) is our full model. First, (a) shows results using only the proposed Sub-Track3R, where the system struggles with large ATE. Adding (b) the point-based loop closure (*Loop_s*) demonstrates that the loop closure significantly improves tracking performance. Comparing (a) and (c) shows that combining Sub-Track3R with our compact Gaussian mapper greatly reduces ATE and enables novel view rendering. The (c)-(d) comparison further highlights the benefits of the point-based loop closure for overall performance. Evaluating (e) versus (g) demonstrates the effectiveness of our intra-submap registration, while (f)-(g) shows the improvements brought by global bundle adjustment. Finally, com-

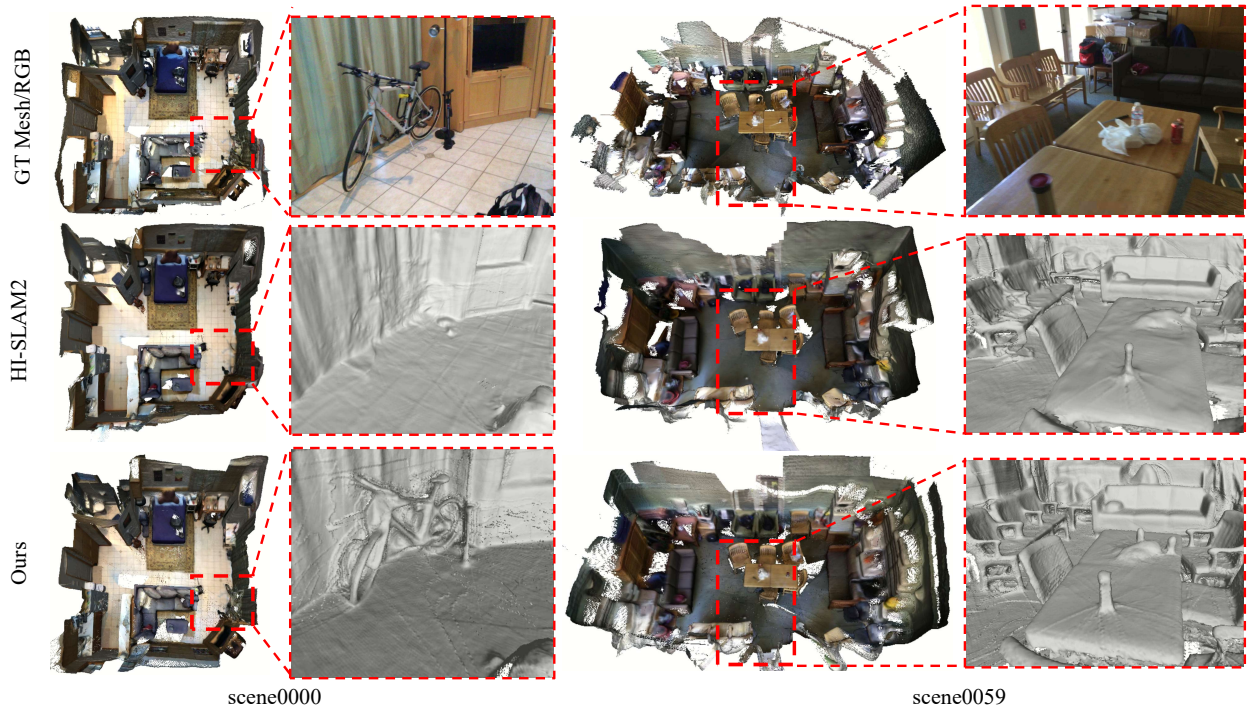

Figure 5. **Qualitative Comparison of Reconstructed Meshes on Scannet-v2 [4].** We compare our reconstructed meshes with the Gaussian-based SLAM method HI-SLAM2 [42]. Our method successfully captures fine scene details, such as the bicycle in scene *0000* and the chair's armrests in scene *0059*, demonstrating superior geometric fidelity and reconstruction quality.

Method	Acc. ↓	Complet. ↓	Chamfer ↓
DROID-SLAM [31]	0.141	0.048	0.094
MASt3R-SLAM [23]	0.068	0.045	0.056
VGGT-SLAM [20]	0.052	0.058	0.055
SING3R-SLAM (Ours)	0.056	0.057	0.057

Table 3. **Quantitative Comparison of 3D Reconstruction on 7-Scenes.** Our method maintains a strong balance across all metrics. Although incomplete ground-truth clouds penalize our more complete reconstructions, our results remain competitive and are supported by strong qualitative comparisons.

paring (d) and (g) illustrates the impact of our bidirectional loop closure, emphasizing the strong synergy between the point map representation and the Gaussian global map.

Performance Analisys. Table 5 reports runtime and memory usage on the scene *0059* of the ScanNet-v2 [4]. SING3R-SLAM achieves tracking time on par with MASt3R-SLAM while producing an additional global Gaussian map. Notably, our global BA can run in parallel with tracking and mapping, minimizing overhead. SING3R-SLAM also maintains a compact 7 MB map size, much smaller than MASt3R-SLAM (110 MB), demonstrating strong efficiency and scalability. The reported map size reflects the final scene representation: for MASt3R-SLAM we use its point maps (its only representation), while for HI-SLAM2 and SING3R-SLAM we report the size of the Gaussian model.

#	Loop	Loop_s	Intra.	Mapper	GBA	ATE ↓	PSNR ↑
(a)						104.15	-
(b)		✓				34.25	-
(c)			✓	✓	✓	24.54	20.17
(d)		✓	✓	✓	✓	11.21	26.72
(e)	✓			✓	✓	12.25	25.66
(f)	✓		✓	✓		9.39	26.43
(g)	✓		✓	✓	✓	**7.20**	**29.44**

Table 4. **Ablations on Key Components of SING3R-SLAM.** "Intra." represents intra-submap registration. (g) is our full model.

Method	Tracking	Mapping	GBA	Map Size
MASt3R-SLAM [23]	**5min**	-	-	110 MB
HI-SLAM2 [42]	8min	12min	10min	9 MB
SING3R-SLAM (Ours)	**5min**	**10min**	**8min**	**7 MB**

Table 5. **Performance Analysis.**

5. Conclusion

In this work, we presented SING3R-SLAM, a submap-based dense RGB SLAM framework that integrates local 3D reconstruction with a globally consistent Gaussian map. Our Sub-Track3R module provides accurate local geometry, while the Gaussian mapper performs multi-view global optimization to maintain a compact and coherent scene representation. Extensive experiments demonstrate that SING3R-SLAM achieves state-of-the-art tracking, reconstruction, and rendering performance with high efficiency, showing the benefits of unifying 3D reconstruction priors with Gaussian representations for dense monocular SLAM.

References

[1] Elena Alegret, Kunyi Li, Sen Wang, Siyun Liang, Michael Niemeyer, Stefano Gasperini, Nassir Navab, and Federico Tombari. Gala: Guided attention with language alignment for open vocabulary gaussian splatting. *arXiv preprint arXiv:2508.14278*, 2025. 1, 2

[2] Oliver Bimber and Ramesh Raskar. Modern approaches to augmented reality. In *Acm siggraph 2006 courses*, pages 1–es. 2006. 1

[3] Timothy Chen, Ola Shorinwa, Joseph Bruno, Aiden Swann, Javier Yu, Weijia Zeng, Keiko Nagami, Philip Dames, and Mac Schwager. Splat-nav: Safe real-time robot navigation in gaussian splatting maps. *IEEE Transactions on Robotics*, 2025. 1

[4] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017. 6, 7, 8, 2, 3, 5

[5] Bardienus Pieter Duisterhof, Lojze Zust, Philippe Weinzaepfel, Vincent Leroy, Yohann Cabon, and Jerome Revaud. Mast3r-sfm: a fully-integrated solution for unconstrained structure-from-motion. In *2025 International Conference on 3D Vision (3DV)*, pages 1–10. IEEE, 2025. 2

[6] Ainaz Eftekhar, Alexander Sax, Jitendra Malik, and Amir Zamir. Omnidata: A scalable pipeline for making multi-task mid-level vision datasets from 3d scans. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10786–10796, 2021. 2

[7] Xinli Guo, Weidong Zhang, Ruonan Liu, Peng Han, and Hongtian Chen. Motiongs: Compact gaussian splatting slam by motion filter. In *2024 7th International Conference on Robotics, Control and Automation Engineering (RCAE)*, pages 685–692. IEEE, 2024. 3

[8] Berthold KP Horn and Brian G Schunck. Determining optical flow. *Artificial intelligence*, 17(1-3):185–203, 1981. 2

[9] Luo Juan and Oubong Gwun. A comparison of sift, pca-sift and surf. *International Journal of Image Processing (IJIP)*, 3(4):143–152, 2009. 2

[10] Nikhil Keetha, Jay Karhade, Krishna Murthy Jatavallabhula, Gengshan Yang, Sebastian Scherer, Deva Ramanan, and Jonathon Luiten. Splatam: Splat track & map 3d gaussians for dense rgb-d slam. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21357–21366, 2024. 2, 3, 5, 6

[11] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023. 2, 3, 4

[12] Xiaohan Lei, Min Wang, Wengang Zhou, and Houqiang Li. Gaussnav: Gaussian splatting for visual navigation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025. 1

[13] Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3d with mast3r. In *European Conference on Computer Vision*, pages 71–91. Springer, 2024. 1, 2

[14] Kunyi Li, Michael Niemeyer, Nassir Navab, and Federico Tombari. Dns-slam: Dense neural semantic-informed slam. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7839–7846. IEEE, 2024. 1

[15] Renwu Li, Wenjing Ke, Dong Li, Lu Tian, and Emad Barsoum. Monogs++: Fast and accurate monocular rgb gaussian slam. *arXiv preprint arXiv:2504.02437*, 2025. 3

[16] Yanyan Li, Youxu Fang, Zunjie Zhu, Kunyi Li, Yong Ding, and Federico Tombari. 4d gaussian splatting slam. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 25019–25028, 2025. 1, 2

[17] Siyun Liang, Sen Wang, Kunyi Li, Michael Niemeyer, Stefano Gasperini, Nassir Navab, and Federico Tombari. Supergseg: Open-vocabulary 3d segmentation with structured super-gaussians. *arXiv preprint arXiv:2412.10231*, 2024. 1, 2

[18] Yuzheng Liu, Siyan Dong, Shuzhe Wang, Yingda Yin, Yanchao Yang, Qingnan Fan, and Baoquan Chen. Slam3r: Real-time dense scene reconstruction from monocular rgb videos. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 16651–16662, 2025. 2, 3

[19] Yuxing Long, Wenzhe Cai, Hongcheng Wang, Guanqi Zhan, and Hao Dong. Instructnav: Zero-shot system for generic instruction navigation in unexplored environment. *arXiv preprint arXiv:2406.04882*, 2024. 1

[20] Dominic Maggio, Hyungtae Lim, and Luca Carlone. Vggt-slam: Dense rgb slam optimized on the sl (4) manifold. *arXiv preprint arXiv:2505.12549*, 2025. 2, 3, 4, 5, 6, 7, 8, 1

[21] Hidenobu Matsuki, Riku Murai, Paul HJ Kelly, and Andrew J Davison. Gaussian splatting slam. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18039–18048, 2024. 1, 2, 3, 6

[22] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. Orb-slam: A versatile and accurate monocular slam system. *IEEE transactions on robotics*, 31(5):1147–1163, 2015. 1, 2

[23] Riku Murai, Eric Dexheimer, and Andrew J Davison. Mast3r-slam: Real-time dense slam with 3d reconstruction priors. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 16695–16705, 2025. 2, 3, 5, 6, 7, 8, 1

[24] Michael Niemeyer, Fabian Manhardt, Marie-Julie Rakotosaona, Christina Tsalicoglou Michael Oechsle, Keisuke Tateno, Jonathan T Barron, and Federico Tombari. Learning neural exposure fields for view synthesis. In *NeurIPS*, 2025. 1

[25] Songyou Peng, Kyle Genova, Chiyu Jiang, Andrea Tagliasacchi, Marc Pollefeys, Thomas Funkhouser, et al. Openscene: 3d scene understanding with open vocabularies. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 815–824, 2023. 2

[26] Minghan Qin, Wanhua Li, Jiawei Zhou, Haoqian Wang, and Hanspeter Pfister. Langsplat: 3d language gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20051–20060, 2024. 2

[27] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *2011 International conference on computer vision*, pages 2564–2571. Ieee, 2011. 2

[28] Erik Sandström, Ganlin Zhang, Keisuke Tateno, Michael Oechsle, Michael Niemeyer, Youmin Zhang, Manthan Patel, Luc Van Gool, Martin Oswald, and Federico Tombari. Splat-slam: Globally optimized rgb-only slam with 3d gaussians. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 1680–1691, 2025. 2, 3, 6

[29] Dieter Schmalstieg and Tobias Hollerer. *Augmented reality: principles and practice*. Addison-Wesley Professional, 2016. 1

[30] Jamie Shotton, Ben Glocker, Christopher Zach, Shahram Izadi, Antonio Criminisi, and Andrew Fitzgibbon. Scene coordinate regression forests for camera relocalization in rgb-d images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2930–2937, 2013. 6, 7, 2, 4

[31] Zachary Teed and Jia Deng. Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras. *Advances in neural information processing systems*, 34:16558–16569, 2021. 1, 2, 3, 8

[32] Sebastian Thrun. Probabilistic robotics. *Communications of the ACM*, 45(3):52–57, 2002. 1

[33] Hengyi Wang and Lourdes Agapito. 3d reconstruction with spatial memory. In *2025 International Conference on 3D Vision (3DV)*, pages 78–89. IEEE, 2025. 2

[34] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 5294–5306, 2025. 1, 2, 3

[35] Qianqian Wang, Yifei Zhang, Aleksander Holynski, Alexei A Efros, and Angjoo Kanazawa. Continuous 3d perception model with persistent state. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 10510–10522, 2025. 2, 6, 1

[36] 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. 1, 2, 3

[37] 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*, pages 10371–10381, 2024. 2

[38] Vladimir Yugay, Yue Li, Theo Gevers, and Martin R Oswald. Gaussian-slam: Photo-realistic dense slam with gaussian splatting. *arXiv preprint arXiv:2312.10070*, 2023. 3, 6

[39] Baowen Zhang, Chuan Fang, Rakesh Shrestha, Yixun Liang, Xiaoxiao Long, and Ping Tan. Rade-gs: Rasterizing depth in gaussian splatting. *arXiv preprint arXiv:2406.01467*, 2024. 4, 6

[40] Ganlin Zhang, Shenhan Qian, Xi Wang, and Daniel Cremers. Vista-slam: Visual slam with symmetric two-view association. *arXiv preprint arXiv:2509.01584*, 2025. 3, 6

[41] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 6

[42] 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. 2, 3, 4, 5, 6, 7, 8, 1

[43] Jianhao Zheng, Zihan Zhu, Valentin Bieri, Marc Pollefeys, Songyou Peng, and Iro Armeni. Wildgs-slam: Monocular gaussian splatting slam in dynamic environments. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 11461–11471, 2025. 3

SING3R-SLAM: Submap-based Indoor Monocular Gaussian SLAM with 3D Reconstruction Priors

Supplementary Material

A. Implementation Details

A.1. Hyperparameters

In the following, we report implementation details and hyperparameters used for our method.

Mapper. We set the learning rate of camera poses as 0.001 for rotation and 0.005 for translation. For Gaussian training, we set the position learning rate as 0.0005, feature learning rate as 0.005, opacity learning rate as 0.05, scaling learning rate as 0.001, rotation learning rate as 0.001. For Intra-Submap Registration, we minimize Eq. 3 with 50 iterations per frame. For map update, we minimize Eq. 6 with 20 iterations. During global bundle adjustment, we densify the Gaussians ever 200 iterations.

Bidirectional Loop Closure. We set the learning rate for the rigid transformation $\mathcal{T}$ as 0.005 and the optimization iteration as 1000.

B. Keyframe Selection

Different from prior SLAM systems [20, 23, 42], we additionally leverage the 3D encoder as a motion filter for keyframe selection. Given an input image C, the 3D encoder (CUT3R [35] in our implementation) encodes the $H \times W \times 3$ image into a compact $H/16 \times W/16 \times 1024$ feature map, which is further decoded into a point map and camera pose.

For each incoming frame, we compute the motion change with respect to the latest keyframe using its encoded features. Concretely, we flatten the feature maps and measure their similarity using the patch-level overlap ratio:

$$r = \frac{1}{N} \sum_{i=1}^{N} \mathbf{1}\big[\max_j \langle f_i^{(t)}, f_j^{(k)} \rangle > \beta\big], \tag{11}$$

where $\mathbf{1}[\cdot]$ is an indicator function, $f_i^{(t)}$ is a feature patch from the current frame, $f_j^{(k)}$ is from the last keyframe, and $\beta = 0.7$ is a similarity threshold. If the overlap ratio r is lower that a patch-level overlap threshold $th = 0.7$, that indicates substantial viewpoint or scene-motion change, prompting us to select the current frame as a new keyframe.

This feature-based motion filtering provides a robust and lightweight criterion, enabling stable keyframe selection even under challenging lighting, textureless regions, and dynamic motions.

C. Covisibility Graph and Loop Detection

Given previous point maps $\{X_i\}_{i=0}^{M}$ and poses $\{T_i^{world}\}_{i=0}^{M}$, and the incoming frame after Sub-Track3R (Sec. 3.1) with point map X_{M+1} and pose T_{M+1}, we reproject the current point map to all previous frames and compute the overlap ratio. If this ratio exceeds 0.3, we add a covisibility edge between the two frames. Despite its simplicity, this intuitive method is fast and effective.

During loop detection, for each incoming frame, we collect all covisible frames whose temporal distance exceeds 10 frames as loop candidates. For each candidate, we compute a loop score that jointly considers geometric overlap and feature similarity. Specifically, we evaluate the bidirectional pointmap overlap r_{pts}, which ensures spatially accurate loop detection by measuring the consistency of reconstructed 3D geometry. In addition, we compute the feature similarity r_{feat} between image features using Eq. 11, which helps avoid false positives caused by transient occlusions or ambiguous geometric structures. The final loop score is computed as

$$\text{score} = 0.7\, r_{pts} + 0.3\, r_{feat}.$$

If the score of a candidate exceeds a threshold (0.5 in our implementation), the incoming frame is recognized as closing a loop, and we invoke the Bidirectional Loop Closure module (Sec. 3.3). This combined geometric–appearance criterion yields robust loop detection even in cluttered or visually repetitive indoor environments.

D. Exposure Compensation

To improve rendering quality and handle illumination variations across different frames, we apply exposure compensation as in previous method [42] to the rendered RGB images:

$$\hat{\mathbf{C}} = \mathbf{A} \times \bar{\mathbf{C}} + \mathbf{b}, \tag{12}$$

where $\bar{\mathbf{C}}$ is the RGB image rendered from the Gaussian rasterizer, $\hat{\mathbf{C}}$ is the exposure-corrected image, $\mathbf{A} \in \mathbb{R}^{3\times3}$ is a linear color transformation matrix, and $\mathbf{b} \in \mathbb{R}^3$ is a bias vector. This simple linear adjustment compensates for global illumination differences between frames, improving color consistency in the rendered sequence.

E. More Experiment Results

E.1. Pose Estimation

Table 6 reports the Absolute Trajectory Error (ATE) on six ScanNet-v2 [4] sequences. SING3R-SLAM achieves competitive pose accuracy, performing on par with state-of-the-art Gaussian-based SLAM systems. Although HI-SLAM2 [42] attains the lowest average ATE, it relies on external tracking from DROID-SLAM [31], resulting in a decoupled tracking–mapping pipeline that limits its reconstruction quality. In contrast, our method tightly integrates 3D reconstruction with a globally optimized Gaussian map, enabling both stable tracking and high-fidelity geometry.

Compared with reconstruction-based MASt3R-SLAM [23] and VGGT-SLAM [34], which show large pose errors on several scenes, SING3R-SLAM provides consistently strong pose estimates while achieving substantially better 3D reconstructions. This demonstrates that our unified design yields advantages in both trajectory accuracy and geometric quality, outperforming methods focused solely on either reconstruction or tracking.

E.2. Rendering

We present the average rendering performance in Table 2 of the main paper. For completeness, Table 7 further reports per-scene rendering results on ScanNet-v2 [4], showing that our advantages hold consistently across individual sequences. Novel view synthesis has become an important metric for assessing 3D reconstruction quality, as it captures both geometric accuracy and multi-view consistency. Our method achieves the best performance among RGB-D and RGB-only Gaussian SLAM approaches, benefiting from a compact and globally consistent Gaussian map that supports high-fidelity rendering.

E.3. Geometry Reconstruction

We provide additional qualitative results on 7-Scenes [30] and ScanNet-v2 [4]. In Fig. 6, we present the reconstructed point clouds. For our approach, we visualize both the dense point cloud and the Gaussian points. Across all scenes, our reconstructions are significantly cleaner than those of prior methods, exhibiting no redundant floating points and thus delivering much clearer scene geometry. In Fig. 7, we compare reconstructed meshes with HI-SLAM2 [42]. Our method recovers substantially finer structures and geometric details, demonstrating the advantage of our globally consistent Gaussian representation.

A video comparison is also provided through an accompanying supplementary webpage included in the submission, where all video results are embedded for convenient viewing.

Method	Avg.	00	59	106	169	181	207
MASt3R-SLAM [23]	7.95	6.96	8.48	9.53	8.34	7.16	7.20
VGGT-SLAM [20]	19.3	12.79	10.21	39.89	22.38	12.63	17.97
MonoGS [21]	122.7	149.2	96.8	155.5	140.3	92.6	101.9
Splat-SLAM [28]	7.66	5.57	9.11	7.09	8.26	8.39	7.53
HI-SLAM2 [42]	7.16	5.82	7.30	6.80	8.25	7.41	7.40
SING3R-SLAM (Ours)	7.41	5.70	7.20	6.75	8.02	8.47	8.31

Table 6. **Quantitative Comparison of Camera Tracking Accuracy on ScanNet-v2.** Our SING3R-SLAM achieves competitive trajectory accuracy compared with state-of-the-art SLAM systems, despite being primarily designed for globally consistent 3D reconstruction rather than pose-only optimization. While methods such as HI-SLAM2 [42] benefit from an external tracker (DROID-SLAM [31]) and decoupled mapping, their reconstructed geometry remains coarse and incomplete. In contrast, our reconstruction-driven SLAM tightly couples 3D geometry with Gaussian mapping, enabling superior structural recovery and richer scene details, while maintaining strong pose accuracy across all sequences.

	Method	Metric	Avg.	0000	0059	0106	0169	0181	0207
RGB-D input	Spla TAM[10]	PSNR ↑	20.42	18.70	20.91	19.84	22.16	22.01	18.90
		SSIM ↑	0.78	0.71	0.79	0.81	0.78	0.82	0.75
		LPIPS ↓	0.38	0.48	0.32	0.32	0.34	0.42	0.41
	Gaussian -SLAM[38]	PSNR ↑	27.67	28.54	26.21	26.26	28.60	27.79	28.63
		SSIM ↑	0.92	0.93	0.93	0.93	0.92	0.92	0.91
		LPIPS ↓	0.25	0.27	0.21	0.22	0.23	0.28	0.29
RGB input	Splat- SLAM[28]	PSNR ↑	29.48	28.68	27.69	27.70	31.14	31.15	30.49
		SSIM ↑	0.85	0.83	0.87	0.86	0.87	0.84	0.84
		LPIPS ↓	0.18	0.19	0.15	0.18	0.15	0.23	0.19
	HI-SLAM2[42]	PSNR ↑	29.27	28.62	27.22	28.13	31.28	30.37	30.03
		SSIM ↑	0.88	0.85	0.87	0.90	0.90	0.90	0.86
		LPIPS ↓	0.24	0.28	0.23	0.21	0.18	0.25	0.30
	SING3R- SLAM[Ours]	PSNR ↑	30.47	28.97	29.44	29.77	31.53	31.66	31.44
		SSIM ↑	0.89	0.84	0.90	0.91	0.90	0.91	0.87
		LPIPS ↓	0.21	0.26	0.18	0.18	0.17	0.23	0.23

Table 7. **Quantitative Comparison of Per Scene Rendering Performance on ScanNet-v2 [4].**

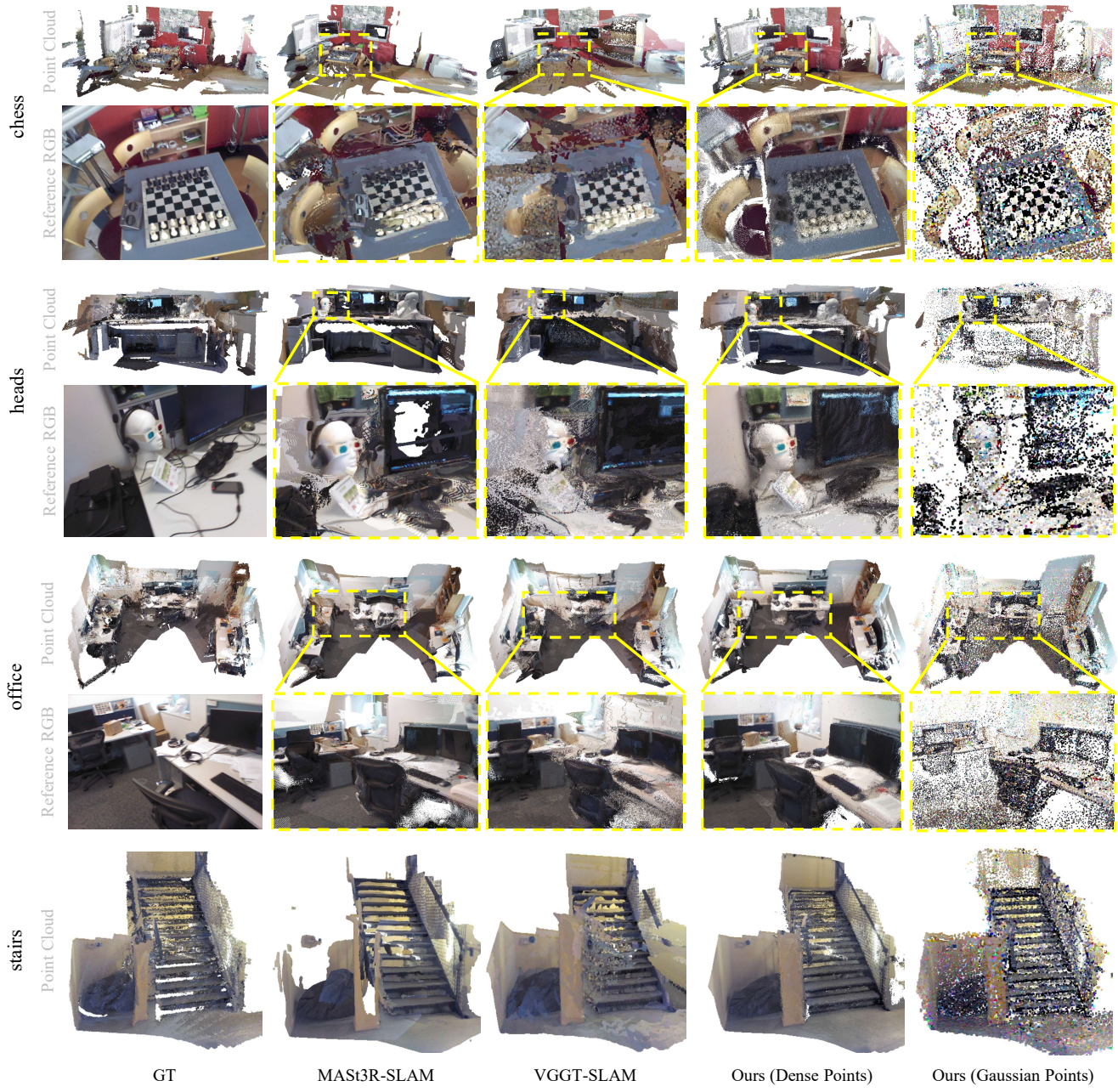

Figure 6. **Qualitative Comparison of Reconstructed Point Clouds on 7-scenes [30].** We show the reconstructed point clouds with zoomed-in views for all methods. Our approach provides a compact Gaussian representation that is much cleaner and captures object geometry in detail, as illustrated in the last column. In contrast, other 3D reconstruction-based methods often produce many redundant points, which degrade visual quality. Moreover, our dense point reconstruction preserves geometric structures more accurately.

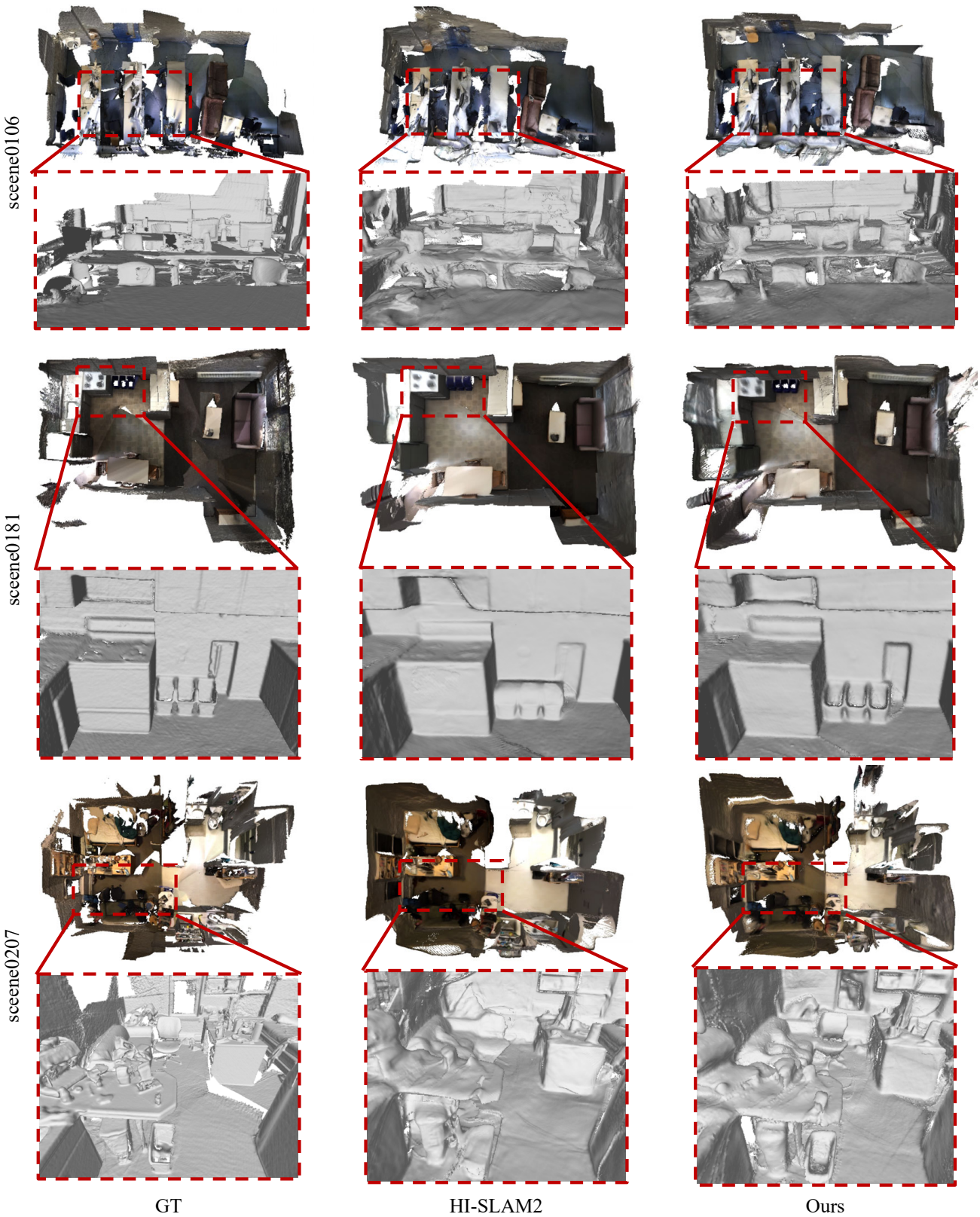

Figure 7. **Qualitative Comparison of Reconstructed Meshes on ScanNet-v2 [4].**