MGS-SLAM: Monocular 3D Gaussian Splatting SLAM with Significance-Guided Pruning

Anonymous cvm submission

Paper ID 435

Abstract

3D Gaussian Splatting demonstrates significant advantages in high-fidelity rendering for *offline* reconstruction of objects or scenes, however, its integration with existing SLAM systems especially for monocular scenario still suffers from limited localization accuracy and poor rendering quality. In this paper, we propose a new monocular 3D Gaussian splatting SLAM, which achieves high fidelity *online* 3DGS-based reconstruction given monocular video input with significance-guided pruning. Our key observation is to maintain structural compactness for the 3D Gaussians optimization, by adaptively pruning the 3D Gaussians under a global significance evaluation from multi-dimensional cues such as visibility, opacity, and volume coefficient. Specifically, we leverage a frame-to-model pipeline that jointly optimizes camera poses and 3D Gaussians within a sliding-window framework, ensuring globally consistent 3D Gaussian representations for high-fidelity rendering with the guidance of significance-guided pruning. Besides, an online monocular depth estimation model is incorporated to extract depth priors from input images, to effectively initialize the 3D Gaussian attributes for better camera tracking. Extensive experiments on Replica and TUM datasets demonstrate that our approach substantially improves both tracking performance and rendering fidelity, achieving state-of-the-art results compared to existing methods.

***Keywords:* Gaussian Splatting, SLAM, Significance-Guided Pruning**

1. Introduction

Visual simultaneous localization and mapping (vSLAM) provides spatial perception capabilities for key applications such as virtual reality (VR), augmented reality (AR), robotics, and unmanned aerial vehicles, demonstrating significant advantages and potential [57, 78, 72, 3, 63]. In recent years, vSLAM has been increasingly extended to diverse application scenarios, leading to enhanced functionalities such as robust operation in dynamic environments [69, 8], semantic-level mapping [58], and high-quality scene reconstruction [17], which have further strengthened its integration with downstream applications [10]. Among them, online scene reconstruction based SLAM [5, 15, 79] aims to simultaneously estimate camera poses and reconstruct 3D scenes that support free-viewpoint, high-fidelity rendering, thereby enabling immersive experiences for applications such as VR and AR [17, 54, 12].

Traditional SLAM methods, which represent scenes using primitives such as point clouds [53], meshes, or voxels, can effectively capture geometric structures but struggle to support high-fidelity rendering [54]. Neural Radiance Fields (NeRF) [35], on the other hand, leverage implicit neural networks to build compact and continuous scene representations, enabling free-viewpoint, photorealistic image synthesis and further enhancing the accuracy of SLAM based scene reconstruction [54, 51]. However, its ray-tracing-based pipeline, which queries volumetric densities and colors at numerous points along pixel-derived rays, is extremely time-consuming and cannot support applications that require real-time rendering. Even with subsequent developments using efficient representations such as octrees [68] or hash tables [38], when integrated with SLAM systems, they often suffer from either low tracking accuracy or poor reconstruction fidelity [33, 56].

In recent years, 3D Gaussian Splatting [18] has significantly improved rendering photorealism and efficiency. It represents scenes using 3D Gaussian ellipsoids, and achieves fast rendering through a differentiable rasterization with adaptive densification. Its integration with SLAM has been extended to various sensing modalities [73], including RGB [34], RGB-D [41], LiDAR [61], and inertial measurement units (IMUs) [20], where the deeply fused sensor data provide richer environmental information, enabling more robust and accurate scene reconstruction. However, monocular SLAM combined with 3D Gaussian Splatting still struggles with inaccurate localization and low reconstruction fidelity.

In this paper, our goal is to accurately reconstruct 3D scenes in a 3D Gaussian splatting SLAM framework, called

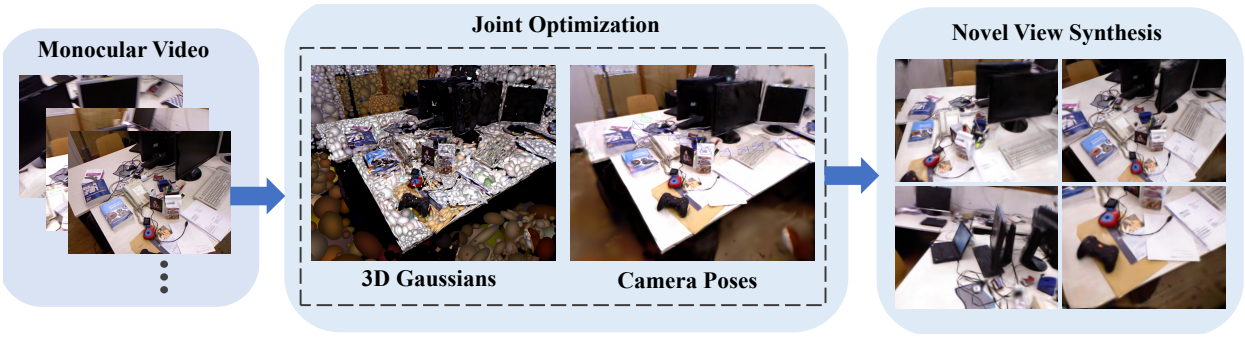

Figure 1. This paper proposes a novel monocular Gaussian Splatting SLAM, i.e., MGS-SLAM. Given streaming monocular video without camera poses, our method jointly optimizes 3D Gaussians and camera poses with a significance-guided pruning strategy, achieving accurate scene reconstruction and high-quality novel view synthesis.

MGS-SLAM, given monocular video input, while enabling real-time, high-fidelity rendering using a novel significance-guided pruning as shown in Figure 1. Unlike previous 3DGS-based SLAM systems that maintain accurate 3D Gaussian reconstruction by fusing rich multi-modal cues, our MGS-SLAM proposes to leverage a significance-guided pruning strategy to retain more significant 3D Gaussians to accurately estimate camera poses, thus enabling high-fidelity online scene reconstruction even using monocular images. Specifically, we introduce a frame-to-model SLAM framework that jointly optimizes camera poses and 3D Gaussian parameters under a photometric loss, employing a sliding window strategy over keyframes. Moreover, we evaluate the significance of each Gaussian by integrating multi-dimensional cues such as visibility count, opacity, and volume coefficient, guiding effective pruning of less significant Gaussians. Besides, we employ an online monocular depth estimation model to obtain depth priors from images, which are then combined with color information to initialize the 3D Gaussian attributes including spatial position, opacity, and color.

Benefiting from the above components, our MGS-SLAM achieves accurate scene reconstruction, ensuring both high-fidelity and real-time rendering performance. Furthermore, MGS-SLAM is evaluated on widely used Replica [49] and TUM [50] datasets, and compared with state-of-the-art monocular scene reconstruction SLAM methods including Photo-SLAM [14], MonoGS [34] and DepthGS [75]. The experimental results indicate that MGS-SLAM achieves state-of-the-art performance in both tracking and rendering performance.

2. Related work

2.1. Scene Reconstruction

Compared with traditional geometry-based reconstruction methods, NeRF provides a continuous implicit representation that jointly models geometry and appearance, enabling high-fidelity novel view synthesis with view-dependent effects [35, 4, 36]. Nevertheless, NeRF's volumetric rendering makes the rendering process extremely time-consuming, which significantly limits its practical applicability. To further enhance NeRF's rendering quality, KiloNeRF [45] and Block-NeRF [52] divide the scene into multiple smaller sub-scenes, each represented individually, while NISB-MAP [60] extends these approaches by introducing overlapping sub-scenes to improve the consistency and stitching quality in intersecting regions.

In recent years, 3D Gaussian Splatting [18] has significantly improved both the rendering fidelity and computational efficiency of NeRF [35] and its variants [68, 38, 45, 52, 60, 37, 1, 43, 55]. Furthermore, some methods have been proposed to achieve more accurate geometric reconstructions [70, 66, 13, 62, 65, 22], among them, GESs [66] represent the scene using a set of 2D opaque surfels with a few 3D Gaussians surrounding them, significantly improving geometric accuracy and enhancing rendering realism; 3DGS-DR [65] leverages the per-pixel reflection gradients generated by deferred shading to bridge the optimization process of neighboring Gaussians, significantly improving non-Lambertian reconstruction quality. In addition, some methods [25, 71, 32, 11] have successfully reconstructed vivid human avatars, expanding the application of scene reconstruction in dynamic contexts [28, 59]. Our work is inspired by those works but focuses on a monocular online reconstruction SLAM using 3D Gaussian splatting.

2.2. Scene Reconstruction SLAM

To support accurate online appearance reconstruction, iMAP [51] first integrates SLAM with NeRF [35], enabling the joint optimization of camera poses and the implicit map. NICE-SLAM [77] and Vox-Fusion [64] adopt a voxel-based representation that stores implicit scene information across the entire environment. Furthermore, several studies have integrated implicit representations with more efficient data structures, including octrees [33], hash ta-

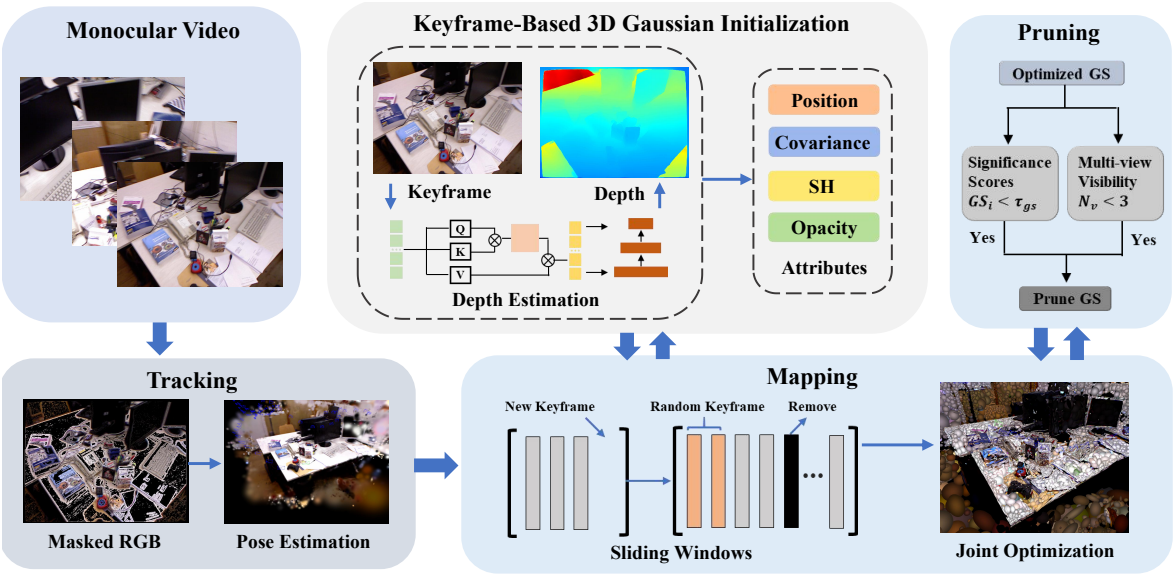

Figure 2. The pipeline of MGS-SLAM. Our method adopts a frame-to-model SLAM framework that jointly optimizes camera poses and 3D Gaussian parameters employing a sliding window strategy (Section 3.1). Moreover, we evaluate the significance of each Gaussian by integrating multi-dimensional cues guiding effective pruning of less significant Gaussians (Section 3.2). Besides, we employ an online monocular depth estimation model to obtain depth priors to efficiently initialize the 3D Gaussian attributes (Section 3.3).

bles [56], tri-planes [16], and point clouds [48], with certain structures even achieving real-time tracking and reconstruction. Moreover, some NeRF-based SLAM methods have incorporated loop closure [30] and submap strategies [6], enabling more accurate camera localization and globally consistent reconstruction, particularly in long-sequence scenarios. In addition, NeRF-based SLAM methods targeting semantic mapping [19], robustness in dynamic scenes [21, 47], and multi-sensor fusion [31] have further expanded its applications.

Recently, some studies integrating 3D Gaussians with SLAM have demonstrated superior rendering speed and reconstruction accuracy compared to NeRF-based SLAM. Among them, Photo-SLAM [14] integrates ORB-SLAM [39] with 3D Gaussian Splatting [18], supporting multiple input modalities, including stereo and RGB-D cameras; RTG-SLAM [41] features a compact Gaussian representation and a highly efficient on-the-fly Gaussian optimization scheme, achieving high-quality reconstruction with low memory cost; GPS-SLAM [42] combines colorized signed distance fields with 3D Gaussians, significantly improving reconstruction speed while maintaining high-quality reconstruction; and MonoGS [34] enables relatively robust tracking and reconstruction. In addition, recent studies have also extended 3D Gaussian Splatting SLAM to dynamic scenes[27, 23] and semantic segmentation [26]. In contrast, our work provides a new significance-guided pruning for 3DGS-based SLAM, which effectively improves the rendering quality especially given monocular inputs.

3. MGS-SLAM

The goal of our MGS-SLAM is to estimate camera poses and reconstruct 3D scenes from monocular input, while achieving real-time, high-fidelity rendering. As shown in Figure 2, we introduce a frame-to-model SLAM framework, which performs joint optimization of camera poses and 3D Gaussians using a sliding-window strategy (Section 3.1). To maintain structural compactness for the 3D Gaussians optimization, the significance of each Gaussian is quantified by integrating multi-dimensional cues, guiding effective pruning of Gaussians (Section 3.2). Additionally, keyframe depths are estimated online and subsequently combined with color information to initialize the attributes of 3D Gaussians (Section 3.3).

3.1. MGS-SLAM Framework

Gaussian Splatting. Our MGS-SLAM is represented using 3D Gaussians, each parameterized with opacity α^i and color c_i. Among them, the α^i serves as a weighting factor in volumetric rendering, while the direction-dependent c_i is modeled via spherical harmonics (SHs) according to the viewing direction. In addition, each 3D Gaussian is characterized by a mean μ_W^i and variance Σ_W^i, which define its spatial position and ellipsoidal shape within the scene, respectively. The final pixel color c_p is derived through volume rendering, where weighted-color contributions from individual Gaussian ellipsoids are computed and

subsequently accumulated, as formulated in Equation (1):

$$c_p = \sum_{i \in N} c_i \alpha_i \prod_{j=1}^{i-1} (1 - \alpha_j), \tag{1}$$

where N denotes the number of 3D Gaussians contributing to the rendering, and the α_i is obtained by decaying α^i via a 2D Gaussian function. Specifically, the 2D Gaussians $\mathcal{N}(\mu_I, \Sigma_I)$ on the image plane are derived from the 3D Gaussians $\mathcal{N}(\mu_W, \Sigma_W)$ through a projective transformation, as shown in Equation (2).

$$\mu_I = \pi(T_{CW} \cdot \mu_W), \Sigma_I = JW\Sigma_W W^T J^T, \tag{2}$$

where π denotes the projection operator, T_{CW} is the estimated camera pose, J represents the Jacobian matrix that provides a linear approximation of the projection transformation, and W denotes the camera rotation matrix. This computation is fully differentiable, allowing for the iterative optimization of the 3D Gaussians' color, scale, rotation, opacity, and position under photometric loss or other relevant constraints.

Tracking. We employ a frame-to-model pipeline for camera pose optimization, where the photometric loss integrates opacity and image gradient cues, as formulated in Equation (3).

$$L_{phot} = E\Big[S(\mathcal{G}, T_{CW}) \odot \big\|I(\mathcal{G}, T_{CW})(M_{fd}) - \bar{I}(M_{fd})\big\|_1\Big], \tag{3}$$

where $S(\mathcal{G}, T_{CW})$ denotes the accumulated opacity of the Gaussians $\mathcal{G}$ from T_{CW}, $I(\mathcal{G}, T_{CW})$ denotes rendered image, $\bar{I}$ is an observed image, and M_{fd} denotes the mask obtained from image gradients, which are computed using a Scharr filter. The image is divided into multiple blocks, and M_{fd} is determined for each block according to its median gradient, as formulated in Equation (4).

$$M_{fd} = \begin{cases} \text{True}, & \text{if} \quad D_j \geq f D_{mi}, \\ \text{False}, & \text{else} \quad D_j < f D_{mi}, \end{cases} \tag{4}$$

where D_j denotes the j-th gradient in the i-th block, D_{mi} denotes the mean gradient of the i-th block, and f is a scaling factor.

Pixel regions exhibiting pronounced color variations are identified by M_{fd}, as they generally correspond to robust texture features that contribute to improved tracking accuracy. In addition, the accumulated opacity s_p, computed as shown in Equation (5), is rendered and subsequently used for adjusting the loss weighting.

$$s_p = 1 - \prod_{i=1}^{N} (1 - \alpha_i). \tag{5}$$

Incorporating the accumulated opacity into pose estimation can mitigate the interference with tracking accuracy caused by insufficient reconstruction. With above loss function, tracking is carried out for It_t iterations, yet the iterations are stopped early whenever the magnitude of the pose update drops below τ_t.

Mapping with Significance-Guided Pruning. In the mapping process, 3D Gaussians are optimized and camera poses are fine-tuned using a sliding-window approach. Specifically, each sliding window comprises the two most recent keyframes along with n additional keyframes whose overlap coefficient (OC) with the current keyframe exceeds a predefined threshold τ_{cko}, where OC is defined as the intersection of 3D Gaussians observed by the two frames divided by the number of Gaussians observed by the frame with fewer observations. A Gaussian is considered observed by a frame if it is used in the rasterization process and its ray's s_p is below 0.5. To dynamically maintain the window, when the window size exceeds its maximum capacity, the keyframe that exhibits large global distance from the nearest keyframe while maintaining high spatial redundancy with neighboring views is removed to preserve a more uniform inter-frame spacing within the keyframe distribution. Specifically, we compute the inverse distance between each frame's pose and those of the other frames in the window, summing these distances to quantify spatial redundancy, with higher values indicating greater redundancy. During optimization, to alleviate the issue of catastrophic forgetting, two keyframes randomly sampled from those outside the local window are included in the optimization process. A photometric loss, as formulated in Equation (6), is used to jointly optimize the 3D Gaussians and p nearest camera poses within the local window.

$$L_{phoms} = E\left[\left\|I(\mathcal{G}, T_{CW}) - \bar{I}\right\|_1\right]. \tag{6}$$

To prevent 3D Gaussians from being abnormally elongated along certain directions due to insufficient viewing constraints, an anisotropy penalty, as formulated in Equation (7), is imposed on their scales. This serves to improve the robustness of 3D Gaussian inference during novel-view synthesis.

$$F_{iso} = E\left[\sum_{i=1}^{|\mathcal{G}|} \|s_{3d} - \overline{s_{3d}}\|_1\right], \tag{7}$$

where s_{3d} denotes the 3 scales of each 3D Gaussian, $\overline{s_{3d}}$ represents the mean value of the 3 scales. Accordingly, the final loss function is formulated as Equation (8):

$$L_{phom} = \sum_{\forall k \in \mathcal{W}} L^k_{phoms} + \lambda_{iso} F_{iso}, \tag{8}$$

where $\mathcal{W}$ represents the set comprising keyframes within the local window together with two randomly selected

keyframes, k denotes the index of a keyframe within $\mathcal{W}$, and λ_{iso} is a weighting factor.

For 3D Gaussian Splatting SLAM, the incorporation of keyframe-based 3D Gaussian initialization facilitates efficient scene reconstruction and enhanced representation of fine-grained details. However, although increasing the number of 3D Gaussian primitives enables the preservation of richer scene information, it simultaneously leads to a substantial increase in parameter dimensionality during mapping and optimization, thereby imposing heavy memory demands and compromising the computational efficiency of the SLAM system. Moreover, unavoidable popping artifacts degrade the fidelity of the reconstructed 3D scene and can substantially impair the tracking performance of frame-to-model methods. Traditional pruning strategies, which evaluate 3D Gaussians based solely on opacity, are insufficient for comprehensively measuring their significance. Consequently, locally detailed or structurally complex regions may lose critical Gaussian primitives, which can adversely affect both tracking stability and overall reconstruction quality. To address this, we propose using a global significance score to quantify the significance of 3D Gaussian primitives and develop a corresponding pruning strategy to integrate it into the backend of visual SLAM framework, following adaptive densification.

3.2. Significance-Guided Pruning.

Global Significance Scores. In the backend, global significance scores are computed for all keyframes within the local window. Specifically, for each keyframe, the 3D Gaussians within the view frustum are projected onto the pixel plane via a rasterizer, and their contributions to pixel rendering are computed. By incorporating the opacity and scale attributes of each Gaussian, the global significance scores are then derived, as formalized in Equation (9).

$$GS_j = \sum_{i=1}^{mhw} 1\left(\mathcal{G}_j, r_i\right) \cdot \alpha_j \cdot \prod_{k=1}^{j-1} \left(1-\alpha_k\right) \cdot \gamma\left(\Sigma_j\right), \quad (9)$$

where m denotes the number of camera frames used for the parameter computation, h denotes the image height, and w denotes the image width, $\mathcal{G}_j$ denotes the j-th Gaussian in the 3D scene, and let $1\left(\mathcal{G}_j, r_i\right)$ indicate whether the j-th Gaussian contributes to the rendering of the i-th pixel, taking a value of 1 if it contributes and 0 otherwise. $\alpha_j \cdot \prod_{k=1}^{j-1}\left(1-\alpha_k\right)$ denotes the weight of the j-th Gaussian contributing to the rendering of the i-th pixel. Since a single Gaussian may participate in the rendering of multiple pixels, all pixels are traversed to accumulate the number of times each primitive contributes. In the computation of global significance scores, a 3D Gaussian is considered more significant if it contributes greater weight across a larger number of rendered pixels. Moreover, since a higher-volume 3D Gaussian generally contributes more to rendering for both the training viewpoints and novel views, it is therefore typically more significant. However, using Gaussian volume alone tends to overemphasize background Gaussians, leading to excessive pruning of Gaussians modeling fine geometry. To address this issue, $\gamma\left(\Sigma_j\right)$ is computed based on the scales of each 3D Gaussian, thereby modulating the influence of its volume on the global significance score robustly. The computation of $\gamma\left(\Sigma_j\right)$ is defined in Equation (10).

$$\gamma\left(\Sigma_j\right) = \left(V_{\text{norm}}\right)^{\beta}, V_{\text{norm}} = \min\left(\frac{V(\Sigma)}{V_{\max 90}}, 1\right), \quad (10)$$

where $V_{norm} \in (0,1]$ denotes the normalized volume of the 3D Gaussian, β is a hyperparameter, $V_{\max 90}$ denotes the volume at the 90th percentile of all 3D Gaussians in the scene, after sorting them in ascending order by volume. In addition, $V(\Sigma) = \frac{4}{3}\pi abc$, where a, b, and c denote the scales of the 3D Gaussian along its three axes.

Preventing Catastrophic Map Forgetting. Global significance scores provide a means to assess the significance of 3D Gaussian primitives; however, using these scores as the sole criterion for pruning can lead to substantial forgetting, since the local window's field of view may not fully capture the global scene. Although this could be mitigated by rendering from all viewpoints rather than only those in the local window to obtain a more globally representative significance measure, such an approach incurs considerable computational overhead. To address this, we directly preserve 3D Gaussians with opacity above a threshold τ_α, which typically encode key structural elements of the scene, as well as primitives with participation counts below a threshold τ_c, which may fall outside the local window's view frustum. Specifically, after excluding the primitives exempted from significance-guided pruning, the remaining Gaussians are sorted by their global significance scores, and pruning is applied according to a pre-defined ratio τ_r. This approach enables efficient integration of significance-guided pruning into our visual SLAM pipeline, mitigating map forgetting while maintaining a compact yet faithful 3D Gaussian splatting scene.

Co-visibility-Guided Pruning. For monocular 3D Gaussian splatting SLAM, the optimized 3D Gaussian primitives often exhibit rendering artifacts due to the lack of geometric priors, which can compromise both tracking and reconstruction fidelity. This issue is especially pronounced for Gaussians inserted by newly selected keyframes, as they are typically constrained by fewer viewpoint observations. To address this, we perform co-visibility-guided pruning within the local window after backend optimization in accordance with multi-view consistency. Specifically, visibility checks are conducted across all keyframes in the local window, and any 3D Gaussian observed by at most two

keyframes and inserted by the three most recent keyframes is pruned. This strategy allows newly inserted Gaussians to more robustly encode the scene while mitigating the impact of multi-view inconsistencies on tracking performance.

3.3. Keyframe-Based 3D Gaussian Initialization

New Keyframe Decision. After obtaining an initial estimate of the camera pose in the frontend, a keyframe selection strategy is employed to determine whether to add new keyframes. Once selected, the new keyframe is incorporated into the current local window, where joint optimization of the 3D Gaussians and camera poses is conducted in the backend. In this process, overly sparse keyframes may lead to insufficient viewpoint coverage, hindering the accurate optimization of 3D Gaussians, whereas overly dense keyframes increase computational overhead and may cause greater accumulation of tracking errors. Keyframe selection is performed by jointly considering frame interval, co-visibility ratio, and inter-frame distance, where co-visibility ratio is defined as the intersection of 3D Gaussians observed by the two frames divided by their union. Specifically, a frame is selected as a keyframe if it satisfies either of the following conditions: (1) the translational distance to its nearest keyframe exceeds $\tau_{tmax}\widehat{D}_m$, where $\widehat{D}_m$ is the median value of the depth map rendered for the current frame, and the frame is separated by at least τ_f frames; (2) the translational distance exceeds $\tau_{tmin}\widehat{D}_m$, the co-visibility ratio exceeds τ_{ck}, and the frame interval is at least τ_f frames. By jointly considering inter-frame distance and co-visibility ratio, the proposed criterion captures both camera motion at the pose estimation level and variations in scene coverage at the reconstruction level, facilitating more accurate and efficient 3D scene reconstruction.

3D Gaussian Initialization. Monocular visual SLAM systems, which rely solely on color information for pose estimation, are prone to tracking drift due to inconsistencies across multiple views. When combined with 3D Gaussian splatting, the lack of geometric priors further limits reconstruction accuracy, as the system has insufficient constraints to reliably recover the 3D scene structure. To address these issues, we propose a depth-enhanced monocular visual SLAM framework, which tightly integrates a monocular depth estimation network into the system frontend. This integration compensates for the inherent lack of depth observations and improves both tracking and mapping performance. Specifically, we employ the DPT [44] monocular depth estimation model pre-trained on the Omnidata scanning datasets, which provides high computational efficiency while producing robust and accurate depth predictions across diverse scenes. To maintain efficiency, depth estimation is performed only on selected keyframes. Using the keyframe images and their corresponding depth maps, along with the preliminarily estimated camera poses and intrinsic parameters, the 3D spatial coordinates and color attributes of pixels are computed via back-projection. Random downsampling is then applied to selected points for the center positions and color attributes of 3D Gaussian primitives, while the scale parameters of each Gaussian are initialized based on distances to their nearest neighbors using a KNN-based approach. By providing accurate depth priors for the initialization of 3D Gaussians, the proposed framework improves both the efficiency and the quality of scene reconstruction.

4. Evaluation

We conduct a comprehensive evaluation of our MGS-SLAM, comprising both qualitative visualizations and quantitative comparisons on real-world and synthetic datasets for 3D Gaussian Splatting–based reconstruction and camera localization. Subsequently, ablation studies are conducted to highlight the contributions of the depth estimation module and the significance-guided Gaussian pruning strategy.

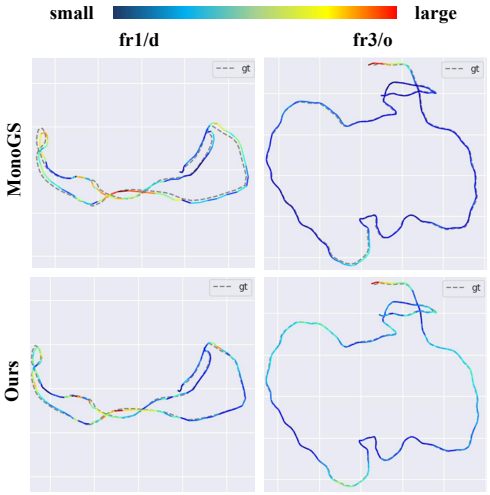

Figure 3. Camera trajectory comparison on TUM datasets.

Table 1. Camera tracking results on TUM datasets. ATE RMSE in cm is reported.

Method	fr1/d	fr2/x	fr3/o	Avg.
DSO [9]	22.40	1.10	9.50	11.00
ORB-SLAM3 [2]	4.33	10.46	123.23	46.01
DepthCov-VO [7]	5.60	1.20	68.80	25.20
DROID-VO [53]	5.20	10.70	7.30	7.73
MonoGS [34]	3.80	4.65	3.60	4.02
DepthGS [75]	9.67	-	-	-
Ours	2.09	3.35	2.10	2.51

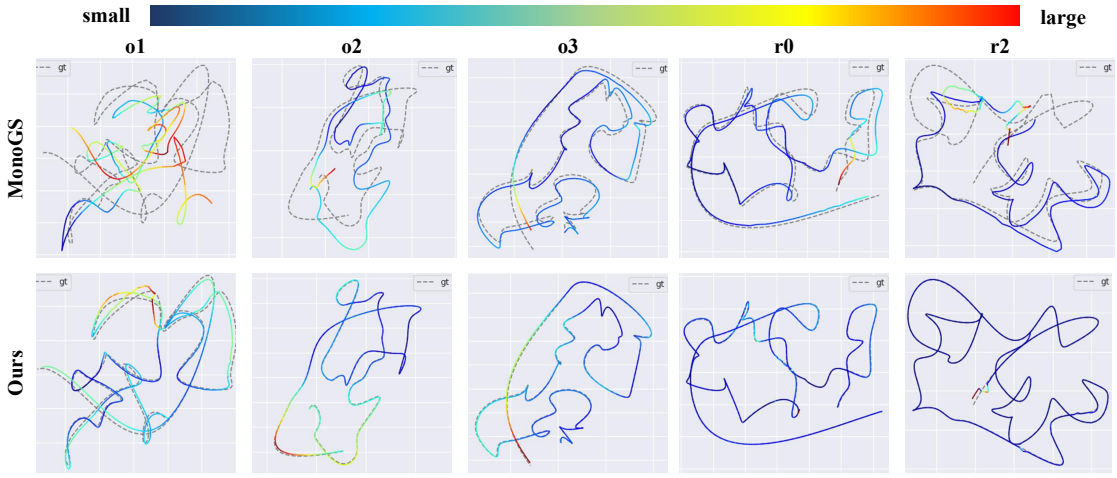

Figure 4. Camera trajectory comparison on the Replica datasets.

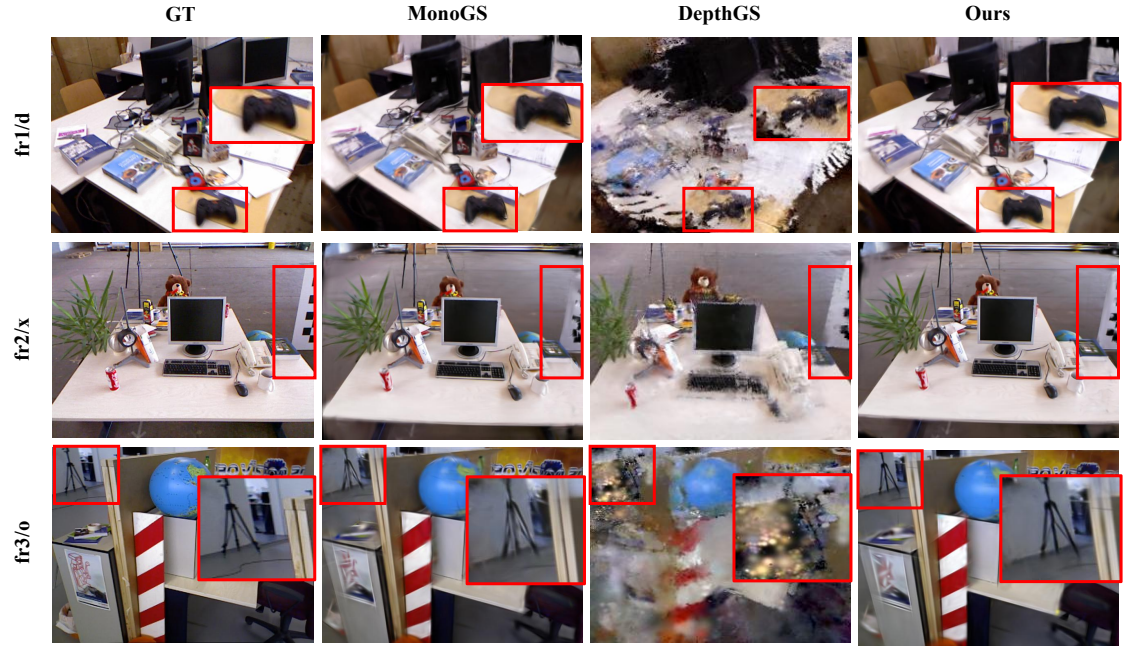

Figure 5. Some qualitative comparison of rendering results between our MGS-SLAM and other methods on TUM datasets.

4.1. Experimental Setup

Datasets. We evaluate MGS-SLAM on the widely adopted real-world TUM [50] datasets, which were captured using a Microsoft Kinect sensor and contain diverse indoor scenes with rich geometric structures, photometric variations, as well as moderate dynamic objects. Furthermore, we conducted experiments on the synthetic Replica [49] datasets, which offer high-resolution textures and extensive camera motion trajectories, enabling a more thorough evaluation of our MGS-SLAM.

Implementation Details. Our experiments are conducted on a computer equipped with an Intel Core i9-12900F 2.4 GHz CPU and a single NVIDIA GeForce RTX 4090 GPU. For tracking, the number of iterations It_t is set to 100 on TUM [50] datasets and 3000 on Replica [49] datasets, with a convergence threshold of $\tau_t = 10^{-4}$ in both cases. Mapping optimization is performed for 150 iterations on TUM [50] and 300 iterations on Replica [49]. Local window maintenance is configured to preserve spatial coverage while controlling redundancy: for TUM [50], we set $\tau_{cko} = 0.3$, $n = 6$, and $p = 3$; for Replica [49], $\tau_{cko} = 0.4$, $n = 8$, and $p = 5$. Keyframe selection was governed by frame interval, co-visibility ratio, and inter-frame distance: for TUM [50], we set $\tau_f = 5$, $\tau_{ck} = 0.9$, $\tau_{tmax} = 0.08$

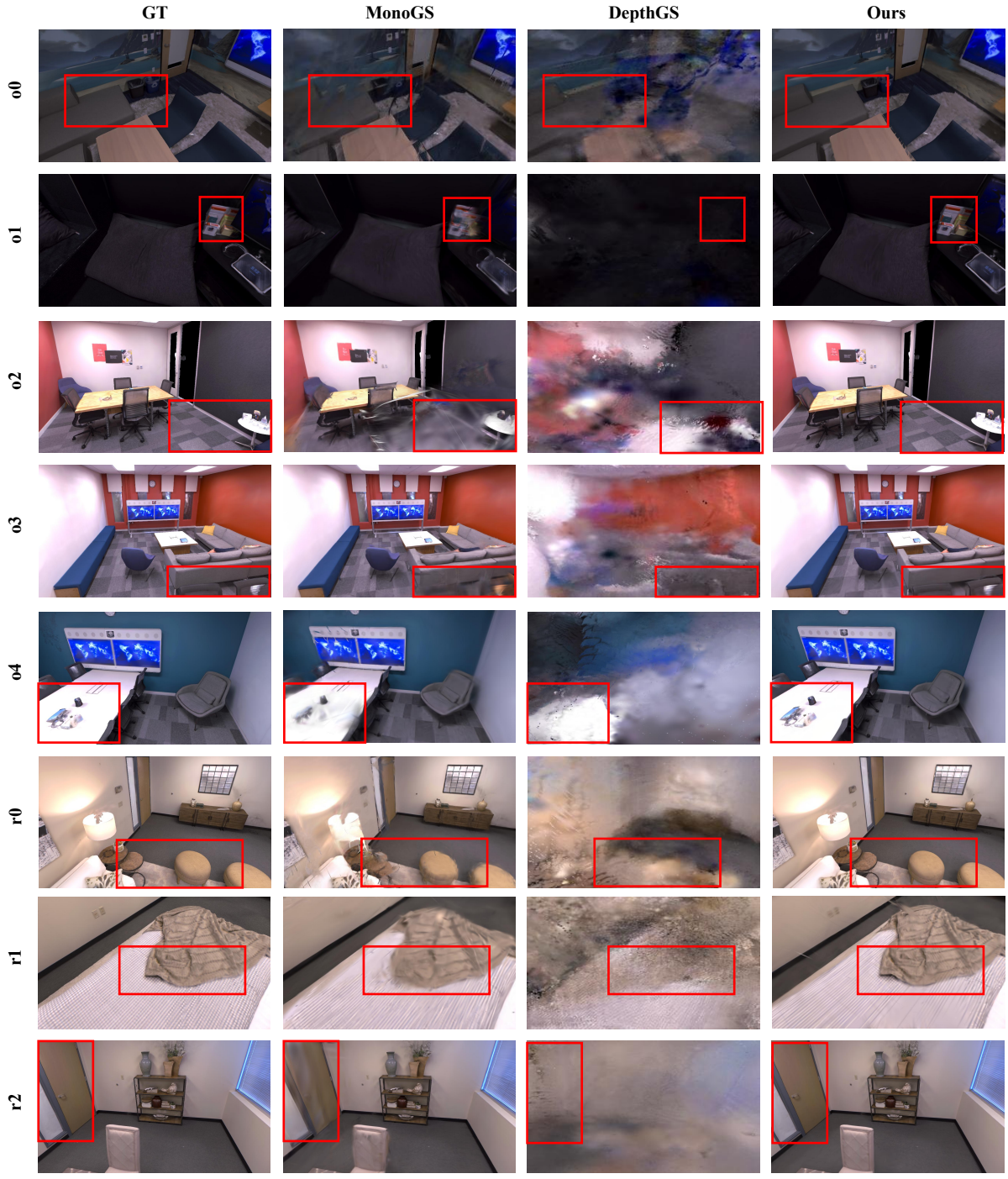

Figure 6. Some qualitative comparison of rendering results between our MGS-SLAM and other methods on Replica datasets.

and $\tau_{tmin} = 0.05$; for Replica [49], $\tau_f = 2$, $\tau_{ck} = 0.975$, $\tau_{tmax} = 0.02$ and $\tau_{tmin} = 0.01$. Pruning of 3D Gaussians was controlled by $\tau_\alpha = 0.7$, $\tau_c = 5$, $\beta = 0.1$, with pruning ratio $\tau_r = 0.85$ for TUM [50] and $\tau_r = 0.95$ for Replica [49]. The mapping loss function incorporated an anisotropy penalty weight $\lambda_{iso} = 10$ for both datasets. These settings ensured stable convergence and high-fidelity reconstruction in our 3D Gaussian Splatting SLAM pipeline.

Metrics. In our experiments, camera pose estimation is quantitatively evaluated using the Root Mean Square Error

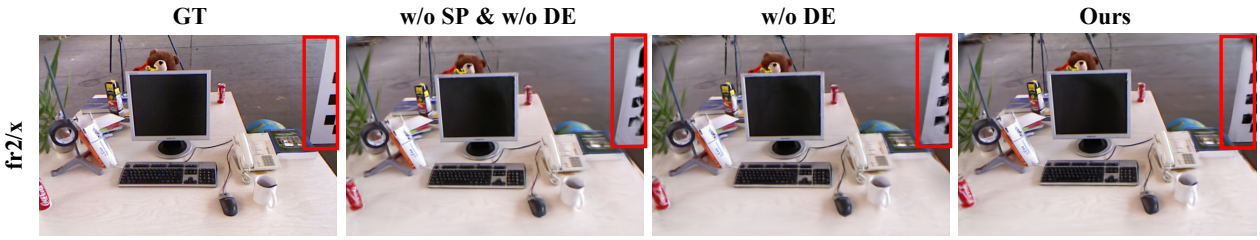

Figure 7. Ablation study of rendering results on the fr2/x sequence of the TUM datasets.

Table 2. Camera tracking results on Replica datasets. ATE RMSE in cm is reported.

Method	o0	o1	o2	o3	o4	r0	r1	r2	Avg.
ORB-SLAM3 [2]	110.21	103.95	65.36	51.15	1.19	51.39	26.38	4.33	51.75
DROID-SLAM [53]	53.27	34.43	119.31	98.09	83.73	103.89	53.15	66.94	76.60
MonoGS [34]	56.54	24.93	41.59	12.29	81.78	10.15	68.27	29.74	40.66
DepthGS [75]	22.00	11.00	-	-	-	-	10.50	-	-
Ours	3.12	1.99	1.37	2.03	11.41	1.21	35.65	1.05	7.23

Table 3. Average rendering performance of our method compared to state-of-the-art methods on TUM and Replica datasets.

Datasets	Metric	GO-SLAM [74]	NICER-SLAM [76]	MoD-SLAM [24]	Photo-SLAM [14]	Mono GS [34]	Q-SLAM [40]	Depth GS[75]	Ours
TUM	PSNR↑	-	-	-	19.68	21.88	-	17.16	22.36
	SSIM↑	-	-	-	0.693	0.731	-	0.654	0.745
	LPIPS↓	-	-	-	0.268	0.330	-	0.487	0.317
Replica	PSNR↑	22.13	25.41	27.31	29.28	29.25	32.49	19.92	34.53
	SSIM↑	0.730	0.827	0.850	0.883	0.890	0.890	0.678	0.932
	LPIPS↓	-	0.191	-	0.139	0.210	0.170	0.478	0.134

Table 4. Quantitative Rendering performance of our method compared to state-of-the-art methods on TUM datasets.

Method	Metric	fr1/d	fr2/x	fr3/o
Photo-SLAM [14]	PSNR↑	18.81	21.35	18.88
	SSIM↑	0.681	0.727	0.672
	LPIPS↓	0.329	0.187	0.289
Mono GS [34]	PSNR↑	21.10	22.43	22.11
	SSIM↑	0.707	0.727	0.760
	LPIPS↓	0.351	0.286	0.353
Depth GS [75]	PSNR↑	17.72	18.30	15.45
	SSIM↑	0.707	0.741	0.515
	LPIPS↓	0.436	0.391	0.635
Ours	PSNR↑	21.39	23.22	22.47
	SSIM↑	0.715	0.750	0.770
	LPIPS↓	0.351	0.258	0.343

(RMSE) of the Absolute Trajectory Error (ATE) computed over selected keyframes. To assess the fidelity of the reconstructed 3D scene, we employ standard photometric metrics, including Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index (SSIM), and Learned Perceptual Image Patch Similarity (LPIPS), following the evaluation protocol established by MonoGS [34].

Baseline Methods. To comprehensively assess the performance of our method, we conducted both quantitative and qualitative comparisons on two benchmark datasets against MonoGS [34] and DepthGS [75], using their publicly available implementations, as they represent the state-of-the-art methods for monocular 3D Gaussian Splatting SLAM. Furthermore, we include comparisons with other advanced methods, such as NICER-SLAM [76], Photo-SLAM [14], and Q-SLAM [40] for a broader evaluation.

4.2. Quantitative Evaluation

Camera Tracking Accuracy. Table 1 reports the tracking results of our method on TUM [50] datasets, together with comparisons to state-of-the-art methods. Our method achieves accurate tracking across three sequences, yielding the best average performance and attaining the highest accuracy on fr1/desk and fr3/office. Although the performance on fr2/xyz is slightly lower than that of DSO [9], our system surpasses MonoGS [34], which is also a Gaussian Splatting based SLAM framework employing a frame-to-model optimization pipeline. Moreover, compared with DepthGS [75], a frame-to-frame SLAM system built upon the 3D Gaussian Splatting representation, our approach achieves significantly higher tracking accuracy on fr1/desk, demonstrating the advantage of our pose estimation strategies.

Table 2 reports the tracking results of our method on the Replica [49] datasets, together with comparisons to state-of-

Table 5. Quantitative rendering performance of our method compared to state-of-the-art methods on Replica datasets.

Method	Metric	o0	o1	o2	o3	o4	r0	r1	r2
NICER-SLAM [76]	PSNR↑	28.54	25.86	21.95	26.13	25.47	25.33	23.92	26.12
	SSIM↑	0.866	0.852	0.820	0.856	0.865	0.751	0.771	0.831
	LPIPS↓	0.172	0.178	0.195	0.162	0.177	0.250	0.215	0.176
Photo-SLAM [14]	PSNR↑	34.42	32.38	28.08	28.06	30.40	26.42	27.08	27.43
	SSIM↑	0.940	0.904	0.900	0.886	0.921	0.787	0.841	0.889
	LPIPS↓	-	0.113	0.131	0.132	0.109	0.221	0.177	0.138
MonoGS [34]	PSNR↑	34.59	34.59	26.59	29.28	27.13	27.95	26.29	27.55
	SSIM↑	0.932	0.933	0.888	0.906	0.902	0.857	0.813	0.888
	LPIPS↓	0.167	0.170	0.240	0.165	0.251	0.168	0.268	0.247
DepthGS [75]	PSNR↑	24.87	26.69	19.23	13.48	14.20	21.54	22.29	17.03
	SSIM↑	0.796	0.869	0.772	0.421	0.521	0.784	0.753	0.508
	LPIPS↓	0.399	0.306	0.377	0.681	0.612	0.391	0.420	0.638
Ours	PSNR↑	36.71	39.00	35.24	34.39	32.14	32.75	29.02	36.96
	SSIM↑	0.949	0.957	0.941	0.939	0.937	0.928	0.851	0.956
	LPIPS↓	0.131	0.122	0.129	0.111	0.160	0.104	0.215	0.099

Table 6. Timing analysis.

System		Rendering	
Time [s]	FPS	Time [s]	FPS
1119.13	3.04	3.67	926.98

Table 7. Tracking results of ablation study on TUM and Replica datasets. ATE RMSE in cm is reported.

Methods	fr1/d	fr2/x	fr3/o	r0
w/o SP & w/o DE	3.66	4.61	3.58	10.07
w/o SP	4.62	3.27	2.70	2.96
w/o DE	3.09	4.42	3.04	14.11
w/o MP	82.42	3.54	6.45	2.21
Ours (DA3)	2.20	0.88	2.70	1.22
Ours (DPT)	2.09	3.35	2.10	1.21

the-art methods. Our method achieves the highest accuracy on six sequences and the best average performance. The lower performance on the o4 and r1 sequences may be attributed to the challenging camera trajectories, which pose difficulties for joint optimization of camera poses and the Gaussian scene representation, yet the method still outperforms MonoGS [34].

Rendering performance. Table 3 and Table 4 report the rendering performance of our method on the TUM [50] datasets, together with comparisons to state-of-the-art methods. Our MGS-SLAM achieves the highest scores in PSNR and SSIM across all three sequences and is comparable to Photo-SLAM [14] in LPIPS, demonstrating its superior rendering fidelity compared with existing methods. The relatively superior LPIPS performance of Photo-SLAM [14] can be attributed to its multi-scale pyramid representation. It is noteworthy that our method is fully compatible with hierarchical representations and can be extended with a hierarchical 3D Gaussian Splatting strategy [46] to further

Table 8. Rendering performance of ablation study on TUM and Replica datasets.

Methods	Metric	fr1/d	fr2/x	fr3/o	r0
w/o SP & w/o DE	PSNR↑	21.10	22.54	22.30	29.69
	SSIM↑	0.707	0.730	0.766	0.897
	LPIPS↓	0.351	0.284	0.348	0.155
w/o SP	PSNR↑	21.19	22.85	22.42	30.06
	SSIM↑	0.713	0.746	0.767	0.886
	LPIPS↓	0.348	0.266	0.345	0.153
w/o DE	PSNR↑	21.15	22.37	22.39	30.09
	SSIM↑	0.709	0.731	0.768	0.905
	LPIPS↓	0.353	0.282	0.346	0.148
w/o MP	PSNR↑	17.22	23.17	21.82	31.97
	SSIM↑	0.600	0.745	0.756	0.920
	LPIPS↓	0.487	0.259	0.362	0.109
Ours (DA3)	PSNR↑	21.39	24.73	22.49	32.52
	SSIM↑	0.717	0.797	0.769	0.924
	LPIPS↓	0.334	0.212	0.340	0.102
Ours (DPT)	PSNR↑	21.39	23.22	22.47	32.75
	SSIM↑	0.715	0.750	0.770	0.928
	LPIPS↓	0.351	0.258	0.343	0.104

enhance the expressive capacity of the scene model. This direction will be explored in future work.

The rendering performance of our method on the Replica [49] datasets is further reported in Table 3 and Table 5 against other state-of-the-art methods. Our approach achieves nearly the best performance across all sequences on the three rendering metrics and attains the highest average scores, demonstrating its clear advantage in capturing scene details and producing high-fidelity renderings.

Timing Analysis. To quantify the computational performance of our system, we measure the total runtime required to process all frames of the TUM [50] fr2/xyz datasets, and

compute the average processing time by dividing the total runtime by the number of frames. Furthermore, all scene images are rendered sequentially, and the average rendering speed is computed in the same manner. As reported in Table 6, our system supports tracking and reconstruction at 3.04 FPS, while achieving a rendering speed of 926.98 FPS. This demonstrates that our method effectively highlights the rendering efficiency advantages of 3D Gaussian Splatting and enables scene reconstruction at a considerable speed.

4.3. Qualitative Results

Camera Tracking Trajectories. As shown in Figure 3 and Figure 4, we visualize the camera tracking trajectories and compare them with the MonoGS [34], which also uses a frame-to-model pipeline and achieves high tracking accuracy. On TUM datasets [50], our method follows the ground-truth camera trajectory (shown as dashed lines) more closely, demonstrating accurate and robust tracking. Furthermore, on Replica [49] datasets, we provide a more extensive comparison of camera trajectories. Our method consistently achieves accurate tracking across trajectories involving various types of motion.

Qualitative Rendering Comparison. As shown in Figure 5 and Figure 6, we reproduced two state-of-the-art methods, MonoGS [34] and DepthGS [75], using their public implementations, and performed extensive comparisons of their rendering results across all sequences. Our method robustly reconstructs all scenes with high fidelity, capturing richer and more realistic details than these baselines.

4.4. Ablative Analysis

To evaluate the effectiveness of our strategies, we conducted ablation experiments on the TUM [50] and Replica [49] datasets. Specifically, for the variant without significance-guided pruning (w/o SP), 3D Gaussians with opacity lower than 0.7 are pruned. For the ablation of the 3D Gaussian initialization (w/o DE), the depth estimation network was not used; instead, the depth values are directly set to 2 with added random noise in the range of ±0.3. For the ablation of co-visibility-guided pruning (w/o MP), we preserve all newly inserted Gaussians. In addition to the DPT [44] depth estimation network primarily evaluated in this work, we also conducted experiments using DA3 [29] to validate the effectiveness of incorporating a depth estimation network for Gaussian initialization. As shown in Table 7, all of our strategies significantly improve tracking accuracy, and Table 8 further shows improvements in rendering quality. Moreover, we provide a qualitative evaluation of their rendering performance on the fr2/x sequence of the TUM [50] datasets in Figure 7, further demonstrating the effectiveness of our strategies.

4.5. Limitations

While our MGS-SLAM demonstrates strong performance in both camera tracking and 3D scene reconstruction, it still exhibits several limitations. The current frame-to-model incremental reconstruction strategy ensures higher geometric consistency, but incurs increased computational cost for pose estimation compared to frame-to-frame methods. Importantly, our 3D Gaussian initialization and significance-guided pruning strategies are fully compatible with frame-to-frame tracking frameworks, offering potential for future integration to balance efficiency and fidelity. As shown in Figure 8, our rendering results on the ScanNet++ [67] sequence indicate that our MGS-SLAM may be challenged by scenes containing prominent non-Lambertian surfaces, owing to the degradation of multi-view consistency and the inherent limitations of 3D Gaussians in modeling color anisotropy. Incorporating semantic priors or reflectance-aware modeling may improve robustness in such non-ideal scenes and enhance the overall fidelity of the reconstructed 3D Gaussians. We leave these directions for future exploration.

Figure 8. Qualitative rendering results on ScanNet++ datasets, with prominent non-Lambertian regions highlighted.

5. Conclusions

We present MGS-SLAM, a system that leverages a frame-to-model pipeline that jointly optimizes camera poses and 3D Gaussians within a sliding-window framework, ensuring globally consistent 3D Gaussian representations for high-fidelity rendering with the significance-guided pruning. Experimental results on the public datasets demonstrate that our method significantly improves both tracking accuracy and rendering fidelity, outperforming existing monocular SLAM methods. We hope this work will inspire further research on more robust and accurate monocular SLAM systems, broadening their potential applications in real-time 3D perception.

References

[1] C. Bao, Y. Zhang, B. Yang, T. Fan, Z. Yang, H. Bao, G. Zhang, and Z. Cui. Sine: Semantic-driven image-based nerf editing with prior-guided editing field. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20919–20929, 2023. 2

[2] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. Montiel, and J. D. Tardós. Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam. *IEEE transactions on robotics*, 37(6):1874–1890, 2021. 6, 9

[3] D. Chen, N. Wang, R. Xu, W. Xie, H. Bao, and G. Zhang. Rnin-vio: Robust neural inertial navigation aided visual-inertial odometry in challenging scenes. In *2021 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 275–283. IEEE, 2021. 1

[4] Q. Chen, K. Huang, Y. Huo, Q. Wang, W. Zheng, R. Li, and R. Xie. Hr human: Modeling human avatars with triangular mesh and high-resolution textures from videos. In *International Conference on Computational Visual Media*, pages 147–171. Springer, 2025. 2

[5] A. Dai, M. Nießner, M. Zollhöfer, S. Izadi, and C. Theobalt. Bundlefusion: Real-time globally consistent 3d reconstruction using on-the-fly surface reintegration. *ACM Transactions on Graphics (ToG)*, 36(4):1, 2017. 1

[6] T. Deng, G. Shen, T. Qin, J. Wang, W. Zhao, J. Wang, D. Wang, and W. Chen. Plgslam: Progressive neural scene represenation with local to global bundle adjustment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19657–19666, 2024. 3

[7] E. Dexheimer and A. J. Davison. Learning a depth covariance function. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13122–13131, 2023. 6

[8] Z.-J. Du, S.-S. Huang, T.-J. Mu, Q. Zhao, R. R. Martin, and K. Xu. Accurate dynamic slam using crf-based long-term consistency. *IEEE Transactions on Visualization and Computer Graphics*, 28(4):1745–1757, 2020. 1

[9] J. Engel, V. Koltun, and D. Cremers. Direct sparse odometry. *IEEE transactions on pattern analysis and machine intelligence*, 40(3):611–625, 2017. 6, 9

[10] Y. Fan, Q. Zhang, Y. Tang, S. Liu, and H. Han. Blitz-slam: A semantic slam in dynamic environments. *Pattern Recognition*, 121:108225, 2022. 1

[11] Z. Fan, S.-S. Huang, Y. Zhang, D. Shang, J. Zhang, Y. Guo, and H. Huang. Rgavatar: Relightable 4d gaussian avatar from monocular videos. *IEEE Transactions on Visualization and Computer Graphics*, 2025. 2

[12] S. Ha, J. Yeon, and H. Yu. Rgbd gs-icp slam. In *European Conference on Computer Vision*, pages 180–197. Springer, 2024. 1

[13] B. Huang, Z. Yu, A. Chen, A. Geiger, and S. Gao. 2d gaussian splatting for geometrically accurate radiance fields. In *ACM SIGGRAPH 2024 conference papers*, pages 1–11, 2024. 2

[14] H. Huang, L. Li, H. Cheng, and S.-K. Yeung. Photo-slam: Real-time simultaneous localization and photorealistic mapping for monocular stereo and rgb-d cameras. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21584–21593, 2024. 2, 3, 9, 10

[15] S.-S. Huang, H. Chen, J. Huang, H. Fu, and S.-M. Hu. Real-time globally consistent 3d reconstruction with semantic priors. *IEEE transactions on visualization and computer graphics*, 29(4):1977–1991, 2021. 1

[16] M. M. Johari, C. Carta, and F. Fleuret. Eslam: Efficient dense slam system based on hybrid representation of signed distance fields. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 17408–17419, 2023. 3

[17] N. Keetha, J. Karhade, K. M. Jatavallabhula, G. Yang, S. Scherer, D. Ramanan, and J. 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. 1

[18] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023. 1, 2, 3

[19] X. Kong, S. Liu, M. Taher, and A. J. Davison. vmap: Vectorised object mapping for neural field slam. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 952–961, 2023. 3

[20] X. Lang, L. Li, C. Wu, C. Zhao, L. Liu, Y. Liu, J. Lv, and X. Zuo. Gaussian-lic: Real-time photo-realistic slam with gaussian splatting and lidar-inertial-camera fusion. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8500–8507. IEEE, 2025. 1

[21] B. Li, Z. Yan, D. Wu, H. Jiang, and H. Zha. Learn to memorize and to forget: A continual learning perspective of dynamic slam. In *European Conference on Computer Vision*, pages 41–57. Springer, 2024. 3

[22] D. Li, S.-S. Huang, and H. Huang. Mpgs: Multi-plane gaussian splatting for compact scenes rendering. *IEEE Transactions on Visualization and Computer Graphics*, 2025. 2

[23] D. Li, S.-S. Huang, Z. Lu, X. Duan, and H. Huang. St-4dgs: Spatial-temporally consistent 4d gaussian splatting for efficient dynamic scene rendering. In *ACM SIGGRAPH 2024 Conference Papers*, pages 1–11, 2024. 3

[24] H. Li, X. Gu, W. Yuan, L. Yang, Z. Dong, and P. Tan. Dense rgb slam with neural implicit maps. *arXiv preprint arXiv:2301.08930*, 2023. 9

[25] L. Li, Y. Li, Y. Weng, Y. Zheng, and K. Zhou. Rgbavatar: Reduced gaussian blendshapes for online modeling of head avatars. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 10747–10757, 2025. 2

[26] M. Li, S. Liu, H. Zhou, G. Zhu, N. Cheng, T. Deng, and H. Wang. Sgs-slam: Semantic gaussian splatting for neural dense slam. In *European Conference on Computer Vision*, pages 163–179. Springer, 2024. 3

[27] Y. Li, Y. Fang, Z. Zhu, K. Li, Y. Ding, and F. Tombari. 4d gaussian splatting slam. *arXiv preprint arXiv:2503.16710*, 2025. 3

[28] G. Liao, Q. Li, Z. Bao, G. Qiu, and K. Liu. Spc-gs: Gaussian splatting with semantic-prompt consistency for indoor open-world free-view synthesis from sparse inputs. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 11264–11274, 2025. 2

[29] H. Lin, S. Chen, J. Liew, D. Y. Chen, Z. Li, G. Shi, J. Feng, and B. Kang. Depth anything 3: Recovering the visual space from any views. *arXiv preprint arXiv:2511.10647*, 2025. 11

[30] L. Liso, E. Sandström, V. Yugay, L. Van Gool, and M. R. Oswald. Loopy-slam: Dense neural slam with loop closures.

In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 20363–20373, 2024. 3

[31] X. Liu, Y. Li, Y. Teng, H. Bao, G. Zhang, Y. Zhang, and Z. Cui. Multi-modal neural radiance field for monocular dense slam with a light-weight tof sensor. In *Proceedings of the ieee/cvf international conference on computer vision*, pages 1–11, 2023. 3

[32] S. Ma, Y. Weng, T. Shao, and K. Zhou. 3d gaussian blendshapes for head avatar animation. In *ACM SIGGRAPH 2024 Conference Papers*, pages 1–10, 2024. 2

[33] Y. Mao, X. Yu, Z. Zhang, K. Wang, Y. Wang, R. Xiong, and Y. Liao. Ngel-slam: Neural implicit representation-based global consistent low-latency slam system. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6952–6958. IEEE, 2024. 1, 2

[34] H. Matsuki, R. Murai, P. H. Kelly, and A. 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, 9, 10, 11

[35] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 1, 2

[36] T.-J. Mu, H.-X. Chen, J.-X. Cai, and N. Guo. Neural 3d reconstruction from sparse views using geometric priors. *Computational Visual Media*, 9(4):687–697, 2023. 2

[37] S. Mubashshira and K. Desai. Te-nerf: Triplane-enhanced neural radiance field for artifact-free human rendering. In *Proceedings of the Winter Conference on Applications of Computer Vision*, pages 238–247, 2025. 2

[38] T. Müller, A. Evans, C. Schied, and A. Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM transactions on graphics (TOG)*, 41(4):1–15, 2022. 1, 2

[39] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos. Orb-slam: A versatile and accurate monocular slam system. *IEEE transactions on robotics*, 31(5):1147–1163, 2015. 3

[40] C. Peng, C. Xu, Y. Wang, M. Ding, H. Yang, M. Tomizuka, K. Keutzer, M. Pavone, and W. Zhan. Q-slam: Quadric representations for monocular slam. *arXiv preprint arXiv:2403.08125*, 2024. 9

[41] Z. Peng, T. Shao, Y. Liu, J. Zhou, Y. Yang, J. Wang, and K. Zhou. Rtg-slam: Real-time 3d reconstruction at scale using gaussian splatting. In *ACM SIGGRAPH 2024 Conference Papers*, pages 1–11, 2024. 1, 3

[42] Z. Peng, K. Zhou, and T. Shao. Gaussian-plus-sdf slam: High-fidelity 3d reconstruction at 150+ fps. *Computational Visual Media*, 2025. 3

[43] A. Pumarola, E. Corona, G. Pons-Moll, and F. Moreno-Noguer. D-nerf: Neural radiance fields for dynamic scenes. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10318–10327, 2021. 2

[44] R. Ranftl, A. Bochkovskiy, and V. Koltun. Vision transformers for dense prediction. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 12179–12188, 2021. 6, 11

[45] C. Reiser, S. Peng, Y. Liao, and A. Geiger. Kilonerf: Speeding up neural radiance fields with thousands of tiny mlps. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 14335–14345, 2021. 2

[46] K. Ren, L. Jiang, T. Lu, M. Yu, L. Xu, Z. Ni, and B. Dai. Octree-gs: Towards consistent real-time rendering with lod-structured 3d gaussians. *arXiv preprint arXiv:2403.17898*, 2024. 10

[47] C. Ruan, Q. Zang, K. Zhang, and K. Huang. Dn-slam: A visual slam with orb features and nerf mapping in dynamic environments. *IEEE Sensors Journal*, 24(4):5279–5287, 2023. 3

[48] E. Sandström, Y. Li, L. Van Gool, and M. R. Oswald. Point-slam: Dense neural point cloud-based slam. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 18433–18444, 2023. 3

[49] J. Straub, T. Whelan, L. Ma, Y. Chen, E. Wijmans, S. Green, J. J. Engel, R. Mur-Artal, C. Ren, S. Verma, et al. The replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019. 2, 7, 8, 9, 10, 11

[50] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers. A benchmark for the evaluation of rgb-d slam systems. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pages 573–580. IEEE, 2012. 2, 7, 8, 9, 10, 11

[51] E. Sucar, S. Liu, J. Ortiz, and A. J. Davison. imap: Implicit mapping and positioning in real-time. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6229–6238, 2021. 1, 2

[52] M. Tancik, V. Casser, X. Yan, S. Pradhan, B. Mildenhall, P. P. Srinivasan, J. T. Barron, and H. Kretzschmar. Block-nerf: Scalable large scene neural view synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8248–8258, 2022. 2

[53] Z. Teed and J. Deng. Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras. *Advances in neural information processing systems*, 34:16558–16569, 2021. 1, 6, 9

[54] F. Tosi, Y. Zhang, Z. Gong, E. Sandström, S. Mattoccia, M. R. Oswald, and M. Poggi. How nerfs and 3d gaussian splatting are reshaping slam: a survey. *arXiv preprint arXiv:2402.13255*, 4:1, 2024. 1

[55] D. Verbin, P. Hedman, B. Mildenhall, T. Zickler, J. T. Barron, and P. P. Srinivasan. Ref-nerf: Structured view-dependent appearance for neural radiance fields. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5481–5490. IEEE, 2022. 2

[56] H. Wang, J. Wang, and L. Agapito. Co-slam: Joint coordinate and sparse parametric encodings for neural real-time slam. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13293–13302, 2023. 1, 3

[57] J. Wang and Y. Qi. Simultaneous scene-independent camera localization and category-level object pose estimation via multi-level feature fusion. In *2023 IEEE Conference Virtual Reality and 3D User Interfaces (VR)*, pages 254–264. IEEE, 2023. 1

[58] Y. Wang, K. Xu, Y. Tian, and X. Ding. Drg-slam: A semantic rgb-d slam using geometric features for indoor dynamic scene. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1352–1359. IEEE, 2022. 1

[59] G. Wu, T. Yi, J. Fang, L. Xie, X. Zhang, W. Wei, W. Liu, Q. Tian, and X. Wang. 4d gaussian splatting for real-time dynamic scene rendering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 20310–20320, 2024. 2

[60] B. Xiang, Y. Sun, Z. Xie, X. Yang, and Y. Wang. Nisb-map: Scalable mapping with neural implicit spatial block. *IEEE Robotics and Automation Letters*, 8(8):4761–4768, 2023. 2

[61] R. Xiao, W. Liu, Y. Chen, and L. Hu. Liv-gs: Lidar-vision integration for 3d gaussian splatting slam in outdoor environments. *IEEE Robotics and Automation Letters*, 2024. 1

[62] T. Xie, X. Chen, Z. Xu, Y. Xie, Y. Jin, Y. Shen, S. Peng, H. Bao, and X. Zhou. Envgs: Modeling view-dependent appearance with environment gaussian. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 5742–5751, 2025. 2

[63] F. Yan, Z. Li, and Z. Zhou. Robust and efficient edge-based visual odometry. *Computational Visual Media*, 8(3):467–481, 2022. 1

[64] X. Yang, H. Li, H. Zhai, Y. Ming, Y. Liu, and G. Zhang. Vox-fusion: Dense tracking and mapping with voxel-based neural implicit representation. In *2022 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 499–507. IEEE, 2022. 2

[65] K. Ye, Q. Hou, and K. Zhou. 3d gaussian splatting with deferred reflection. In *ACM SIGGRAPH 2024 Conference Papers*, pages 1–10, 2024. 2

[66] K. Ye, T. Shao, and K. Zhou. When gaussian meets surfel: Ultra-fast high-fidelity radiance field rendering. *ACM Transactions on Graphics (TOG)*, 44(4):1–15, 2025. 2

[67] C. Yeshwanth, Y.-C. Liu, M. Nießner, and A. Dai. Scannet++: A high-fidelity dataset of 3d indoor scenes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12–22, 2023. 11

[68] A. Yu, R. Li, M. Tancik, H. Li, R. Ng, and A. Kanazawa. Plenoctrees for real-time rendering of neural radiance fields. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5752–5761, 2021. 1, 2

[69] M.-F. Yu, L. Zhang, W.-F. Wang, and J.-H. Wang. Scp-slam: Accelerating dynaslam with static confidence propagation. In *2023 IEEE Conference Virtual Reality and 3D User Interfaces (VR)*, pages 509–518. IEEE, 2023. 1

[70] Z. Yu, T. Sattler, and A. Geiger. Gaussian opacity fields: Efficient adaptive surface reconstruction in unbounded scenes. *ACM Transactions on Graphics (ToG)*, 43(6):1–13, 2024. 2

[71] Y. Zhan, T. Shao, Y. Yang, and K. Zhou. Real-time high-fidelity gaussian human avatars with position-based interpolation of spatially distributed mlps. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 26297–26307, 2025. 2

[72] G. Zhang, J. Yuan, H. Liu, Z. Peng, C. Li, Z. Wang, and H. Bao. 100-phones: A large vi-slam dataset for augmented reality towards mass deployment on mobile phones. *IEEE Transactions on Visualization and Computer Graphics*, 30(5):2098–2108, 2024. 1

[73] P. Zhang, D. Wang, and H. Lu. Multi-modal visual tracking: Review and experimental comparison. *Computational Visual Media*, 10(2):193–214, 2024. 1

[74] Y. Zhang, F. Tosi, S. Mattoccia, and M. Poggi. Go-slam: Global optimization for consistent 3d instant reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3727–3737, 2023. 9

[75] L. Zhao, X. Xu, Y. Wang, H. Wang, W. Zheng, Y. Tang, H. Yan, and J. Lu. Pseudo depth meets gaussian: A feed-forward rgb slam baseline. *arXiv preprint arXiv:2508.04597*, 2025. 2, 6, 9, 10, 11

[76] Z. Zhu, S. Peng, V. Larsson, Z. Cui, M. R. Oswald, A. Geiger, and M. Pollefeys. Nicer-slam: Neural implicit scene encoding for rgb slam. In *2024 International Conference on 3D Vision (3DV)*, pages 42–52. IEEE, 2024. 9, 10

[77] Z. Zhu, S. Peng, V. Larsson, W. Xu, H. Bao, Z. Cui, M. R. Oswald, and M. Pollefeys. Nice-slam: Neural implicit scalable encoding for slam. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12786–12796, 2022. 2

[78] M. Zins, G. Simon, and M.-O. Berger. Oa-slam: Leveraging objects for camera relocalization in visual slam. In *2022 IEEE international symposium on mixed and augmented reality (ISMAR)*, pages 720–728. IEEE, 2022. 1

[79] Z.-X. Zou, S.-S. Huang, Y.-P. Cao, T.-J. Mu, Y. Shan, H. Fu, and S.-H. Zhang. Gp-recon: Online monocular neural 3d reconstruction with geometric prior. *IEEE Transactions on Visualization and Computer Graphics*, 2024. 1