DynaLite-GS: Lightweight Dynamic 3DGS-SLAM in Resource-Constrained Scenarios via Decoupled Masking and Stochastic Backtracking

Houzhi Zhou[1], Li Zhang[1*], Yifan Duan[2], Yu-an Liu[1], Yujie Chen[1], Yingjie Wang[3], Conghao Huang[1], Yuqi Fan[1]

***Abstract*— Existing dynamic 3DGS SLAM approaches predominantly rely on computationally intensive fine-grained segmentation networks, foundation models, or complex iterative algorithms. These methods feature massive computational loads and slow inference speeds, sacrificing practicality for performance gains, thereby hindering deployment on resource-constrained platforms. To address this, we present DynaLite-GS, a lightweight framework that proves sufficient priors can be obtained by relying solely on a lightweight coarse-grained semantic segmentation network. We propose a Decoupled Masking Strategy (DMS) that combines coarse semantic priors with geometric depth consistency and Gaussian visibility to effectively filter dynamic disturbances. Furthermore, to maintain global consistency without the overhead of traditional Pose Graph Optimization, we design a Stochastic Backtracking Mechanism (STBM) tailored for dynamic scenes, which randomly revisits historical keyframes for joint optimization. Extensive experiments on the TUM and BONN datasets demonstrate that DynaLite-GS outperforms state-of-the-art methods in tracking accuracy while maintaining competitive rendering quality. Quantitatively, our system demonstrates significant efficiency gains: it reduces GPU memory usage by up to 83.3%, decreases the cumulative computational load by up to 57.8%, and suppresses the peak primitive count by up to 61.2%, thereby enabling high-quality reconstruction on consumer-grade hardware.**

I. INTRODUCTION

Simultaneous Localization and Mapping (SLAM) is a core technology enabling autonomous robot navigation and augmented reality. A long-standing challenge in this field is achieving dense scene reconstruction while maintaining real-time performance. Although implicit mapping approaches represented by Neural Radiance Fields [1]–[5] have demonstrated impressive reconstruction quality, their high rendering costs and lack of explicit geometric representation limit system real-time capabilities. In contrast, 3D Gaussian Splatting (3DGS) [6], leveraging its rasterization-based real-time rendering capability and explicit map representation, has rapidly emerged as an ideal map representation for dense SLAM.

Currently, static-assumption-based 3DGS SLAM systems have achieved excellent results in both pose estimation and reconstruction quality [7], [8]. However, when facing dynamic environments, these methods suffer from significant mapping artifacts and pose drift. To address this, integrating semantic segmentation priors [9]–[11], as seen in traditional [12], [13] and NeRF-based SLAM [14], [15], has become a standard practice. Consequently, existing dynamic 3DGS SLAM approaches generally fall into two computationally demanding paradigms. The first paradigm relies heavily on dense data priors [16]–[19], such as integrating computationally intensive external visual odometry [20], dense optical flow, or vision foundation models [21] to extract fine-grained dynamic masks. The second paradigm depends on complex mathematical modeling and backend optimization, such as multi-round iterative algorithms, intricate motion probability models, or complex loop closure frame selection modules [22] to maintain consistency.

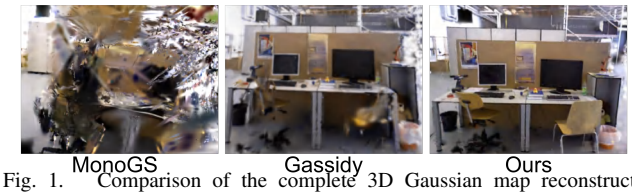

Fig. 1. Comparison of the complete 3D Gaussian map reconstructed on the challenging TUM fr3/wk_xyz sequence. Compared to MonoGS [7] and Gassidy[†] [23], DynaLite-GS more effectively filters out dynamic disturbances, yielding a clean and globally consistent background map. ([†]Rendering of Gassidy is sourced from the original paper.)

Although effective in improving tracking and reconstruction accuracy, both paradigms introduce substantial computational and memory overheads. This reliance on intensive processing severely degrades operational speed, hindering the practical deployment of these frameworks in resource-constrained real-world scenarios such as consumer-grade hardware or battery-powered mobile robots.

To address these deployment constraints, we propose DynaLite-GS, a lightweight dynamic 3DGS SLAM framework tailored for resource-constrained platforms. Adopting a classic front-end and back-end multi-threading architecture, the system bypasses heavy computational dependencies and achieves efficient dynamic reconstruction primarily through two core modules.

First, we introduce the Decoupled Masking Strategy (DMS) to filter dynamic objects. Rather than relying on computationally expensive fine-grained networks, this strategy synergizes extremely lightweight coarse-grained seman-

*Corresponding author: lizhang@hfut.edu.cn
[1]The authors are with Hefei University of Technology, China.
[2]The author is with LYNSENSE, China.
[3]The author is with Nanjing University of Posts and Telecommunications, China.

tic priors with geometric constraints including multi-view depth consistency and Gaussian visibility. Specifically, the generated masks are decoupled according to distinct task requirements and applied precisely to front-end camera tracking as well as back-end pose and Gaussian parameter optimization. Furthermore, we expand the masks via morphological dilation to cover motion blur and introduce a circuit breaker mechanism that dynamically halts operations upon detecting excessive culling. At minimal computational cost, these designs effectively compensate for the coarse segmentation of lightweight priors, strictly safeguarding the geometric integrity of the static background.

Second, we design the Stochastic Backtracking Mechanism (STBM) in the back-end customized for dynamic scenes. This mechanism effectively circumvents the substantial overhead of complex global loop closures while maintaining high-quality global consistency. Embodying the core concept of task decoupling, STBM randomly samples a predefined number of historical keyframes in the background and jointly optimizes them with the main window's loss, naturally incorporating specifically designed backtracking masks and loss functions. Ultimately, this approach significantly suppresses global trajectory and map drift over long-term operation with negligible additional computational overhead.

In summary, our main contributions are:

- We present DynaLite-GS, a lightweight and deployable 3DGS SLAM framework tailored for resource-constrained platforms. By avoiding heavy computational priors, our system achieves high-fidelity reconstruction on consumer-grade hardware.
- We propose a Decoupled Masking Strategy that synergizes coarse semantic priors with geometric depth and visibility constraints. Decoupled for distinct front-end tracking and back-end optimization tasks, and augmented by morphological dilation and a circuit breaker mechanism, it efficiently filters dynamic artifacts while protecting static geometry.
- We design a Stochastic Backtracking Mechanism in the back-end that samples historical keyframes for joint optimization utilizing specific backtracking masks and loss functions. This approach maintains global consistency and mitigates drift without the substantial overhead of complex global loop closure mechanisms.
- Extensive evaluations on TUM and BONN datasets confirm that DynaLite-GS achieves state-of-the-art tracking accuracy and competitive rendering quality. Quantitatively, the system demonstrates exceptional efficiency by reducing GPU memory usage by up to 83.3%, decreasing the cumulative computational load by up to 57.8%, and suppressing the peak primitive count by up to 61.2%.

II. Related Work

A. 3D Gaussian Splatting SLAM

The emergence of 3D Gaussian Splatting has precipitated a paradigm shift in dense SLAM, offering superior real-time rendering and explicit scene representation. Pioneering RGB-D systems, such as SplaTAM [24] and GS-SLAM [8], leverage differentiable rasterization to balance mapping efficiency and accuracy. As the first monocular 3DGS SLAM system, MonoGS [7] achieves high-fidelity reconstruction by deriving analytical Jacobians for pose optimization and incorporating geometric regularization to penalize anisotropy. Subsequent works have further matured the framework, with Splat-SLAM [25] integrating loop closure and global bundle adjustment to ensure global consistency, and GS^{3}LAM [26] formulating the scene as a Gaussian Semantic Field for enhanced understanding. Nevertheless, these state-of-the-art methods are predominantly constrained by a static scene assumption; in dynamic environments, their inability to distinguish moving objects inevitably leads to severe ghosting artifacts and tracking divergence.

B. 3D Gaussian Splatting SLAM in Dynamic Environments

To mitigate the interference caused by dynamic objects, existing Gaussian Splatting SLAM solutions generally fall into two categories. The first category utilizes auxiliary deep networks or external odometry to handle dynamics. For instance, DG-SLAM [16] incorporates a heavy optical flow-based odometry [20] for coarse tracking and utilizes semantic masks for dynamic filtering, while Dy3DGS-SLAM [17] explicitly fuses masks from both optical flow and depth estimation networks. However, despite their superior accuracy, these methods incur prohibitive computational costs and excessive GPU memory consumption, rendering them impractical for deployment on resource-constrained devices. To bypass reliance on heavy networks, the second category attempts to handle dynamics by mining geometric or statistical inconsistencies. WildGS-SLAM [18] employs uncertainty-aware geometric mapping, while MPDG-SLAM [27] introduces a mathematically derived Motion Probability attribute. Similarly, Gassidy [23] iteratively analyzes rendering loss flows, and UP-SLAM [19] proposes a training-free uncertainty estimator [21] to filter dynamic areas. Although reducing dependency on heavy auxiliary components, these methods often involve complex iterative optimizations or statistical modeling, posing challenges for real-time performance on strictly constrained platforms. Additionally, regarding global consistency, DGS-SLAM [22] introduces a loop-aware window selection strategy. However, integrating loop-aware mechanisms into the incremental SLAM pipeline significantly exacerbates the computational burden. To address these challenges, our DynaLite-GS adopts a lightweight decoupled masking strategy to ensure the exclusion of dynamic objects , avoiding reliance on heavy priors or complex iterative optimizations. Furthermore, it introduces a Stochastic Backtracking Mechanism to achieve robust global consistency without heavy loop closure modules.

III. METHOD

The architecture of DynaLite-GS is illustrated in Fig. 2. Our system orchestrates a decoupled mask generation module and a joint optimization strategy with stochastic back-

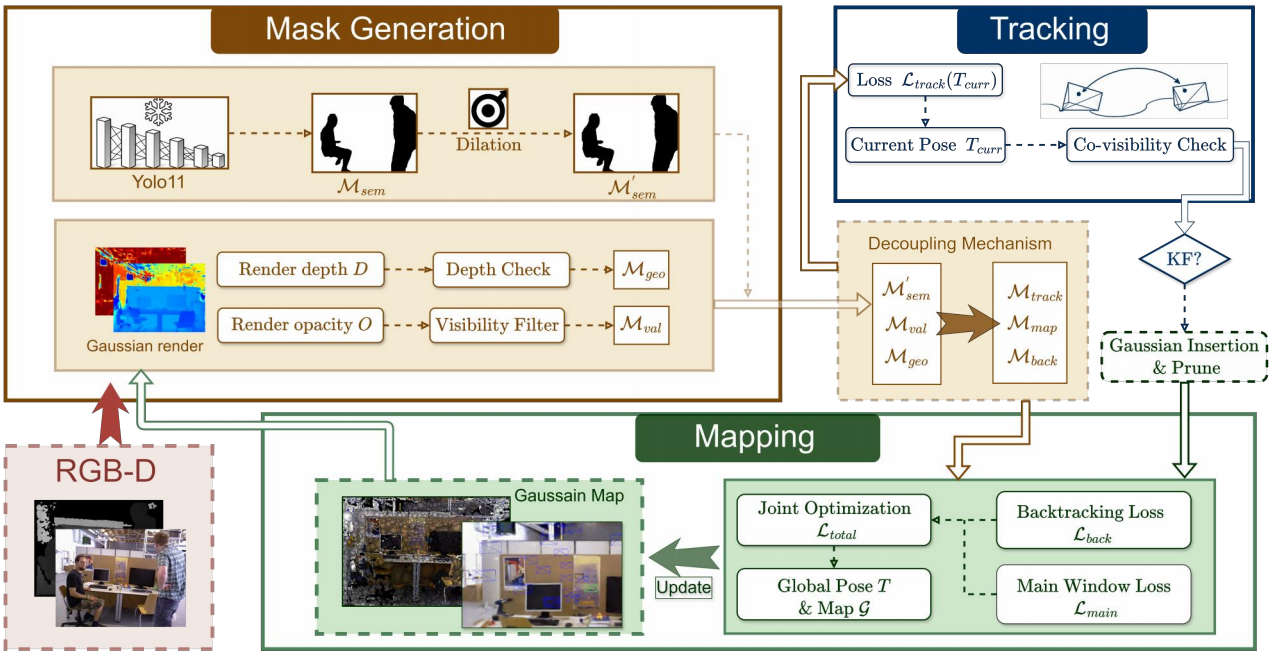

Fig. 2. **System Overview of DynaLite-GS.** DynaLite-GS processes RGB-D streams via a Mask Generation module, which fuses dilated YOLO11 semantics with geometric and visibility cues to synthesize decoupled masks. Subsequently, the Mapping stage maintains global consistency by jointly optimizing the current window and historical keyframes via the backtracking loss $\mathcal{L}_{back}$ to update the Gaussian map $\mathcal{G}$.

tracking to achieve robust trajectory estimation and high-fidelity mapping in complex dynamic environments.

A. 3D Gaussian Splatting Representation

We adopt 3D Gaussian Splatting (3DGS) as the explicit scene representation. The map $\mathcal{G} = \{g_i\}$ consists of primitives defined by mean μ_i, covariance Σ_i, opacity o_i, and spherical harmonics c_i. Given pose T_k, we synthesize color $C_k(\mathbf{u})$, depth $D_k(\mathbf{u})$, and opacity $O_k(\mathbf{u})$ at pixel $\mathbf{u}$ via differentiable α-blending of ordered Gaussians $\mathcal{N}$:

$$\{C_k, D_k, O_k\}(\mathbf{u}) = \sum_{n\in\mathcal{N}} \{c_n, z_n, 1\} \cdot \alpha_n \prod_{m=1}^{n-1} (1-\alpha_m). \quad (1)$$

This formulation enables joint optimization of $\mathcal{G}$ and poses by minimizing residuals between rendered attributes and sensor observations.

B. Decoupled Mask Generation

We propose an efficient Decoupled Masking Strategy integrating visibility, semantic, and multi-view depth consistency to achieve robust tracking and high-quality mapping. We define a binary mask $\mathcal{M}$, where 1 denotes static regions and 0 denotes dynamic regions.

Semantic Prior via Morphological Dilation. We employ a lightweight semantic network for coarse dynamic object segmentation to obtain the initial semantic mask $\mathcal{M}_{sem}$. However, raw predictions often suffer from imprecise boundaries and motion artifacts. To address this, we introduce Morphological Dilation with a kernel K to conservatively expand dynamic boundaries:

$$\mathcal{M}'_{sem} = 1 - ((1 - \mathcal{M}_{sem}) \oplus K), \quad (2)$$

where $\oplus$ denotes the morphological dilation operator. This strategy ensures that ghosting artifacts induced by object motion are completely excluded from the static model.

Geometric Depth Consistency via Sliding Window. To handle generic dynamic objects undefined by the semantic network, we verify geometric depth consistency within a sliding window $\mathcal{W}$. For a pixel $\mathbf{u}$ in frame I_k, we unproject it using rendered depth $D_k(\mathbf{u})$ and warp it to reference frame j via rigid transformation $\mathbf{T}_{jk}$:

$$\mathbf{u}_{k\to j} = \pi\left(\mathbf{T}_{jk} \cdot \pi^{-1}(\mathbf{u}, D_k(\mathbf{u}))\right). \quad (3)$$

We then compute the residual δ against the sensor ground truth depth D_j^{GT}. To robustly distinguish static regions under potential occlusions, we adopt a **Union Strategy**: a pixel is classified as static if it satisfies the consistency constraint in *at least one* reference view:

$$\delta_{k\to j}(\mathbf{u}) = \left|[\mathbf{T}_{jk} \cdot \pi^{-1}(\mathbf{u}, D_k(\mathbf{u}))]_z - D_j^{GT}(\mathbf{u}_{k\to j})\right|, \quad (4)$$

$$\mathcal{M}_{geo}(\mathbf{u}) = \bigvee_{I_j\in\mathcal{W}} \mathbb{I}\left(\delta_{k\to j}(\mathbf{u}) < \tau_{geo}\right), \quad (5)$$

where $\bigvee$ denotes the logical OR operation, and $\mathbb{I}(\cdot)$ represents the indicator function. Additionally, π and π^{-1} denote the camera projection and unprojection functions, respectively, while $[\cdot]_z$ extracts the Z-coordinate of a 3D point.

Opacity-based Visibility Mask. Since dynamic objects fail to form stable Gaussian opacity accumulation, we generate a visibility mask $\mathcal{M}_{vis} = \mathbb{I}(O(\mathbf{u}) > \tau_{vis})$ to filter unstable regions. We adopt a Task-Decoupled Strategy: a strict τ_{track} ensures tracking stability, while a relaxed τ_{map} facilitates map densification.

Algorithm 1 Decoupled Mask Generation Strategy

Input: Current Frame I_k, D_k^{GT}, Gaussian Map $\mathcal{G}$, Keyframe Window $\mathcal{W}$, Semantic Mask $\mathcal{M}_{sem}$, Dilation kernel K, Thresholds $\{\tau_{geo}, \tau_{track}, \tau_{map}, \tau_{break}\}$.
Output: Tracking Mask $\mathcal{M}_{track}$, Mapping Mask $\mathcal{M}_{map}$.

Step 1: Semantic Refinement via Morphological Dilation
$\mathcal{M}'_{sem} \leftarrow 1 - ((1 - \mathcal{M}_{sem}) \oplus K)$
Step 2: Render Depth and Opacity
Render depth D_k and opacity O_k from $\mathcal{G}$ at pose T_{cur}
$\mathcal{M}_{vis}(\tau_{track}) \leftarrow \mathbb{I}(O_k > \tau_{track})$
$\mathcal{M}_{vis}(\tau_{map}) \leftarrow \mathbb{I}(O_k > \tau_{map})$
Step 3: Generate Tracking Mask
$\mathcal{M}_{track} \leftarrow \mathcal{M}'_{sem} \odot \mathcal{M}_{vis}(\tau_{track})$
return $\mathcal{M}_{track}$ to Tracker
Step 4: Geometric Check via Sliding Window
Initialize $\mathcal{M}_{geo}(\mathbf{u}) \leftarrow 0$
for all Reference Frame $I_j \in \mathcal{W}$ **do**
  Warp pixels: $\mathbf{u}_{k\to j} \leftarrow \pi(T_{jk} \cdot \pi^{-1}(\mathbf{u}, D_k))$
  Compute residual: $\delta = |[\mathbf{T}_{jk} \cdot \mathbf{P}_k]_z - D_j^{GT}(\mathbf{u}_{k\to j})|$
  $\mathcal{M}_{geo} \leftarrow \mathcal{M}_{geo} \vee \mathbb{I}(\delta < \tau_{geo})$
end for
Step 5: Mapping Mask with Circuit Breaker
Calculate dynamic ratio: $r_{dyn} \leftarrow \frac{\sum(1-\mathcal{M}_{geo})}{\sum 1}$
if $r_{dyn} > \tau_{break}$ **then**
  $\mathcal{M}_{geo} \leftarrow \mathbf{1}$ {Trigger Circuit Breaker}
end if
$\mathcal{M}_{map} \leftarrow \mathcal{M}'_{sem} \odot \mathcal{M}_{vis}(\tau_{map}) \odot \mathcal{M}_{geo}$
return $\mathcal{M}_{track}, \mathcal{M}_{map}$

Decoupled Fusion with Circuit Breaker Mechanism. We design a Decoupled Fusion Strategy for distinct stages. In the tracking stage, to avoid unreliable geometry before pose convergence, we rely solely on priors:

$$\mathcal{M}_{track} = \mathcal{M}'_{sem} \odot \mathcal{M}_{vis}(\tau_{track}), \tag{6}$$

where $\odot$ denotes the element-wise Hadamard product. In the mapping stage, we incorporate $\mathcal{M}_{geo}$ to excise generic dynamic objects. However, to prevent the erroneous rejection of texture-less static backgrounds misclassified as dynamic due to photometric or depth matching ambiguities, we introduce a Circuit Breaker Mechanism. If the dynamic ratio r_{dyn} exceeds a safety threshold τ_{break}, we trigger a failsafe to bypass the geometric mask (Alg. 1):

$$\mathcal{M}_{map} = \mathcal{M}'_{sem} \odot \mathcal{M}_{vis}(\tau_{map}) \odot \begin{cases} 1, & \text{if } r_{dyn} > \tau_{break} \\ \mathcal{M}_{geo}, & \text{otherwise} \end{cases}. \tag{7}$$

C. Optimization Objective

We jointly optimize poses $T \in SE(3)$ and map $\mathcal{G}$ using stage-specific weighted residuals. In the following formulations, we denote the sensor ground truth color and depth images as I^{GT} and D^{GT}, respectively.

1) Tracking Stage: Fixing $\mathcal{G}$, we optimize the current pose T_{cur}. The per-pixel tracking residual $\mathcal{E}_{track}$ measures the error between the rendered attributes and the ground truth:

$$\mathcal{E}_{track}(\mathbf{u}) = \alpha_1 \|C_{cur}(\mathbf{u}) - I_{cur}^{GT}(\mathbf{u})\|_1 + \alpha_2 \|D_{cur}(\mathbf{u}) - D_{cur}^{GT}(\mathbf{u})\|_1. \tag{8}$$

The loss aggregates residuals over the valid pixel set Ω defined by the mask:

$$\mathcal{L}_{track}(T_{cur}) = \sum_{\mathbf{u}\in\Omega} \mathcal{M}_{track}(\mathbf{u}) \cdot \mathcal{E}_{track}(\mathbf{u}). \tag{9}$$

2) Joint Mapping and Stochastic Backtracking: To guarantee local accuracy and global consistency, we fuse main window optimization with a Stochastic Backtracking Mechanism.

a) Main Window Optimization: For frames in $\mathcal{W}$, we apply the strict $\mathcal{M}_{map}$. The joint residual $\mathcal{E}_{map}$ is:

$$\mathcal{E}_{map}(\mathbf{u}, k) = \alpha_1 \|C_k(\mathbf{u}) - I_k^{GT}(\mathbf{u})\|_1 + \alpha_2 \|D_k(\mathbf{u}) - D_k^{GT}(\mathbf{u})\|_1. \tag{10}$$

The objective function is formulated as:

$$\mathcal{L}_{main}(T_{\mathcal{W}}, \mathcal{G}) = \sum_{k\in\mathcal{W}} \sum_{\mathbf{u}\in\Omega_k} \mathcal{M}_{map}^{(k)}(\mathbf{u}) \cdot \mathcal{E}_{map}(\mathbf{u}, k). \tag{11}$$

b) Isotropic Regularization: Dynamic occlusions frequently yield insufficient multi-view constraints. This scarcity induces geometric ambiguity, causing primitives to degenerate into elongated artifacts aligned with viewing rays—analogous to the instability observed in MonoGS [7] for static monocular settings. To mitigate this ill-posed geometry, we employ an isotropic regularization term to penalize excessive anisotropy:

$$\mathcal{L}_{iso}(\mathcal{G}) = \sum_{i=1}^{|\mathcal{G}|} \|s_i - \bar{s}_i \cdot \mathbf{1}\|_1, \tag{12}$$

where s_i denotes the scaling vector of the i-th primitive and $\bar{s}_i$ is its mean value.

c) Stochastic Backtracking: To mitigate catastrophic forgetting and recover static textures erroneously discarded by strict geometric checks, we sample historical keyframes $\mathcal{K}_{rand}$. Crucially, we employ a relaxed backtracking mask $\mathcal{M}_{back}$ that excludes geometric constraints. This strategy enables the system to re-integrate these previously rejected regions into the optimization process, ensuring that static details misclassified due to view dependency are effectively restored. Consequently, this lightweight backtracking strategy tailored for dynamic scenes maintains superior global consistency. The backtracking mask is defined as:

$$\mathcal{M}_{back}(\mathbf{u}) = \mathcal{M}'_{sem}(\mathbf{u}) \odot \mathcal{M}_{vis}(\mathbf{u}|\tau_{map}). \tag{13}$$

To maintain consistency with the main mapping stage, the backtracking residual $\mathcal{E}_{back}(\mathbf{u}, j)$ follows the identical formulation as $\mathcal{E}_{map}$ (Eq. 10). The backtracking loss is thus computed as:

$$\mathcal{L}_{back}(T_{\mathcal{K}_{rand}}, \mathcal{G}) = \sum_{j\in\mathcal{K}_{rand}} \sum_{\mathbf{u}\in\Omega_j} \mathcal{M}_{back}^{(j)}(\mathbf{u}) \cdot \mathcal{E}_{back}(\mathbf{u}, j). \tag{14}$$

Finally, the total loss function for the mapping stage is defined as:

$$\mathcal{L}_{total} = \mathcal{L}_{main} + \lambda_{back}\mathcal{L}_{back} + \lambda_{iso}\mathcal{L}_{iso}. \tag{15}$$

IV. EXPERIMENTS

A. Experiment Setup

Datasets. We evaluate our system on the TUM RGB-D [28] and BONN RGB-D [29] datasets to demonstrate robustness and accuracy in dynamic environments. The former features rich textures and typical dynamic pedestrian movements, serving to test the system's ability to suppress common disturbances. In contrast, the latter poses significant challenges for tracking and mapping due to the presence of texture-less regions and numerous undefined dynamic objects, such as moving boxes and balloons. Both datasets were captured using handheld devices with aggressive camera motion, fully testing the practical performance of the SLAM system.

Metrics. For tracking accuracy, we employ the Root Mean Square Error (RMSE) of the Absolute Trajectory Error (ATE) to measure global consistency and the Standard Deviation (STD) to assess trajectory stability against dynamic disturbances. For rendering quality assessment, we report the Peak Signal-to-Noise Ratio (PSNR) to evaluate pixel-wise photometric accuracy, the Structural Similarity Index Measure (SSIM) to assess structural fidelity, and the Learned Perceptual Image Patch Similarity (LPIPS) to quantify perceptual realism aligned with human vision.Additionally, we record the average GPU memory usage (GB) to evaluate the memory efficiency of the system on resource-constrained platforms.

Implementation Details. DynaLite-GS is implemented in PyTorch. To assess deployment feasibility on resource-constrained platforms, all experiments are conducted on a hardware setup equipped with an NVIDIA RTX 5060 GPU (8GB) and an AMD Ryzen 5 9600X CPU, running on Ubuntu 20.04. To achieve efficient semantic priors on this platform, we employ the lightweight YOLO11 [11] model. Key hyperparameters specific to our proposed modules are configured as follows: the main sliding window size is set to $|\mathcal{W}| = 8$, and the stochastic backtracking window size is $|\mathcal{K}_{rand}| = 4$; the geometric depth consistency threshold is $\tau_{geo} = 0.15$ m; the circuit breaker safety ratio is $\tau_{break} = 0.5$; the isotropic regularization weight is $\lambda_{iso} = 10$; and the backtracking loss weight is $\lambda_{back} = 1$. For the optimization objective, the balancing factors for the photometric and geometric terms are set to $\alpha_1 = 0.95$ and $\alpha_2 = 0.05$, respectively.

Baselines. To establish a comprehensive performance benchmark, we compare our method against a range of state-of-the-art baselines across different paradigms. These include prominent Gaussian Splatting SLAM works optimized for dynamic scenes: Gassidy [23], DGS-SLAM [22], DG-SLAM [16], Dy3DGS [17]and MPDG-SLAM [27]. Furthermore, we include the static Gaussian baseline MonoGS [7], neural implicit SLAM methods RoDyn-SLAM [3], Co-SLAM [5], and ESLAM [4], as well as the standard geometric method ORB-SLAM3 [30].

Fig. 3. Qualitative mapping results in dynamic environments. The third column presents magnified details of the regions highlighted by red dashed boxes in the rendered images, demonstrating the high-fidelity reconstruction of the static background after dynamic object removal.

B. Tracking Accuracy Evaluation

We evaluate localization performance against state-of-the-art baselines. As reported in Table I, our method demonstrates superior tracking accuracy on the texture-rich and highly dynamic TUM RGB-D dataset. Notably, on the challenging fr3/wk_xyz sequence, DynaLite-GS reduces the ATE to 1.5 cm, achieving a 57% improvement over the second-best baseline. Although Gassidy [23] relies on the complex iterative analysis of rendering loss streams, and DGS-SLAM [22] depends on heavy loop-aware frame selection logic to maintain global consistency, our system achieves this without relying on such high-overhead components. This result compellingly confirms that relying solely on our robust DMS driven by dilated semantic priors and the STBM is sufficient to ensure high-precision global trajectory estimation and achieve state-of-the-art accuracy.

C. Gaussian Splatting Mapping Quality Evaluation

Fig. 3 presents a visual comparison between rendered maps, ground truth images, and occluded static details. Furthermore, Fig. 4 provides a qualitative comparison against state-of-the-art baselines. Qualitatively, even under challenges like undefined dynamic objects (e.g., balloons, laptops), dense crowds, or rapid motion blur, our system ef-

TABLE I: Tracking Accuracy Evaluation on TUM RGB-D Dataset (Unit: cm). Best results are **bolded**, and second-best are underlined. "-" indicates unavailable results.

Sequence	ORB-SLAM3		ESLAM		Co-SLAM		RoDyn-SLAM		MonoGS		Gassidy		DGS-SLAM		Dy3DGS		**DynaLite-GS**	
	ATE	STD	ATE	STD	ATE	STD	ATE	STD	ATE	STD	ATE	STD	ATE	STD	ATE	STD	ATE	STD
fr3/wk_xyz	28.1	12.2	45.7	28.5	51.8	25.3	8.3	5.5	37.2	9.9	3.5	1.6	4.1	2.2	5.8	-	**1.5**	**0.8**
fr3/wk_hf	30.5	9.0	60.8	27.9	105.1	42.0	5.6	2.8	60.0	20.7	3.7	1.9	5.5	2.8	7.0	-	**3.3**	**1.8**
fr3/st_hf	2.6	1.6	3.6	1.6	4.7	2.2	4.4	2.2	7.4	5.4	**2.4**	**1.4**	4.1	1.6	3.4	-	3.9	2.1
fr3/wk_st	2.0	1.1	93.6	20.7	49.5	10.8	1.7	0.9	8.4	4.1	0.6	0.3	0.6	**0.2**	6.5	-	**0.4**	**0.2**
Average	15.8	6.0	50.9	19.7	52.8	20.1	5.0	2.9	28.3	10.0	2.6	1.3	3.6	1.7	5.7	-	**2.3**	**1.2**

TABLE II: Tracking Accuracy Evaluation on BONN Dataset (Unit: cm). Best results are **bolded**, and second-best are underlined. "-" indicates unavailable results.

Sequence	ORB-SLAM3		ESLAM		Co-SLAM		RoDyn-SLAM		MonoGS		Gassidy		DGS-SLAM		**DynaLite-GS**	
	ATE	STD	ATE	STD	ATE	STD	ATE	STD	ATE	STD	ATE	STD	ATE	STD	ATE	STD
balloon	5.8	2.8	22.6	12.2	28.8	9.6	7.9	2.7	33.2	16.4	**2.6**	0.8	2.9	0.8	2.9	**0.6**
balloon2	17.7	8.6	36.2	19.9	20.6	8.1	11.5	6.1	26.5	14.2	7.6	3.4	6.0	2.8	**5.4**	**2.5**
ps_track	70.7	32.6	48.0	18.7	61.0	22.2	14.5	4.6	63.2	29.0	10.3	4.4	9.8	4.1	**9.5**	**3.6**
ball_track	**3.1**	**1.6**	12.4	6.6	38.3	17.4	13.3	4.7	4.3	2.2	-	-	5.6	2.8	5.6	2.6
Average	24.3	11.4	29.8	14.4	37.2	14.3	11.8	4.5	31.8	15.5	-	-	6.1	2.6	**5.9**	**2.3**

TABLE III: Rendering Quality Evaluation on BONN Dataset. We report PSNR↑, SSIM↑, and LPIPS↓ metrics. Best results are **bolded**, and second-best are underlined.

Method	balloon			balloon2			ps_track			ps_track2			Average		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
RoDyn-SLAM	15.75	0.592	0.397	16.68	0.609	0.417	17.96	0.828	0.325	19.40	0.659	0.340	17.45	0.672	0.370
MonoGS	17.70	0.714	0.480	**19.40**	0.748	0.366	18.90	0.731	0.396	20.00	0.756	0.378	19.00	0.737	0.405
DG-SLAM	17.84	0.721	0.367	18.60	**0.857**	0.435	**20.57**	**0.891**	**0.201**	19.79	0.752	**0.310**	19.20	**0.805**	**0.328**
DynaLite-GS	**19.08**	**0.802**	**0.345**	18.66	0.771	**0.349**	19.94	0.787	0.337	**20.46**	**0.783**	0.327	**19.54**	0.786	0.340

TABLE IV: Ablation Study on Key Components. **Bold** indicates the best performance.

Configuration	Avg. on TUM		Avg. on BONN	
	ATE	STD	ATE	STD
Ours	**2.1**	**1.0**	**6.8**	**2.7**
w/o Dilation	3.8	2.6	8.8	3.8
w/o Visibility Mask	3.8	2.3	9.4	3.6
w/o Decoupling	7.8	4.8	21.9	8.2
w/o Stoch. Backtrack	4.5	2.2	8.9	4.2

fectively suppresses ghosting artifacts and produces a clean scene representation. Compared to the baselines, our method demonstrates highly competitive performance in both overall reconstruction quality and the recovery of occluded background details.

Quantitative results in Table III validate this conclusion. Notably, even when compared to methods like DG-SLAM [16], which incorporates a computation-heavy, optical flow-driven visual odometry, our model exhibits on-par rendering quality. Despite prioritizing computational efficiency, our method maintains highly competitive fidelity across PSNR, SSIM, and LPIPS metrics, successfully preventing the severe visual distortion typical of dynamic interference.

D. Ablation Study

To evaluate the impact of individual components on tracking accuracy and mapping quality, we conducted extensive ablation studies on the TUM and BONN datasets. We established the following variants: (1) **w/o Dilation**: disabling the morphological dilation of the semantic mask; (2) **w/o Visibility Mask**: removing the opacity-based visibility mask; (3) **w/o Decoupling**: applying the tracking mask uniformly across all stages, thereby abandoning task-specific decoupling; and (4) **w/o Stoch. Backtrack**: removing the stochastic backtracking module, retaining only the main window optimization.

Quantitative results in Table IV demonstrate the contribution of each module to tracking accuracy. Specifically, semantic dilation effectively covers missed regions at segmentation boundaries, while the visibility check filters out unstable Gaussians with low opacity, preventing tracking drift caused by floating noise. Crucially, the absence of task decoupling forces the system to apply a uniformly strict dynamic mask across all stages. This causes severe over-rejection of static background elements, driving the system into "feature starvation". Consequently, the camera lacks sufficient geometric constraints during pose estimation, which inevitably induces severe tracking drift. Furthermore, the stochastic backtracking mechanism significantly enhances global pose consistency in dynamic scenes. The degradation in performance across all ablated variants confirms that the full system integration is essential for optimal tracking accuracy. Regarding mapping quality, qualitative comparisons in Fig. 5 show that the full method achieves the best performance in both overall map quality and the reconstruction of static details.Specifically, semantic dilation prevents dynamic texture leakage by correcting object boundaries, while the visibility check effectively removes floating

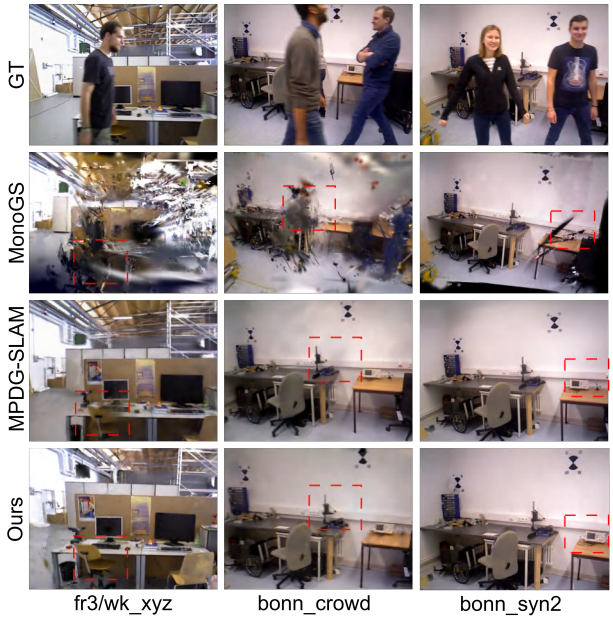

Fig. 4. Qualitative comparison of mapping quality against baseline methods. (†As the source code of MPDG-SLAM [27] is unavailable, its rendering results are directly sourced from the original paper. To ensure a fair comparison, the evaluated frames for the other methods are strictly aligned with those used in MPDG-SLAM.)

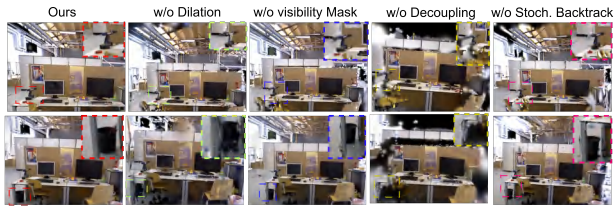

Fig. 5. Qualitative Evaluation on Ablation Study.

artifacts.Crucially, from a global perspective, the absence of task decoupling erroneously eliminates a massive number of valid static map points. Applying the overly strict tracking mask during the mapping phase severely hinders Gaussian optimization, producing extensive unconverged regions and prominent map holes.Consequently, our decoupled method achieves clean and complete background reconstruction even under severe dynamic interference.Meanwhile, the stochastic backtracking mechanism improves the global consistency of the Gaussian map, further validating its necessity for high-quality mapping.

E. Efficiency Analysis

To comprehensively evaluate the system's computational efficiency and memory management over extended operations, we specifically selected long sequences from the TUM dataset for this experiment. Conventional short sequences often fail to adequately expose the memory accumulation issues inherent in standard 3DGS pipelines. In complex dynamic environments, these standard pipelines typically accumulate redundant primitives to fit photometric errors, drastically increasing memory and computational overhead.

TABLE V: System Efficiency Comparison. **Bold** indicates the best performance.

Method	**GPU**	**CPU**	**GMU (GB)** ↓
RoDyn-SLAM	RTX 3090 Ti	Core i7	4.00
DG-SLAM	RTX 3090 Ti	-	9.00
MPDG-SLAM	RTX 3090	Core i7	6.40
DynaLite-GS	RTX 5060	Ryzen 5	**1.50**

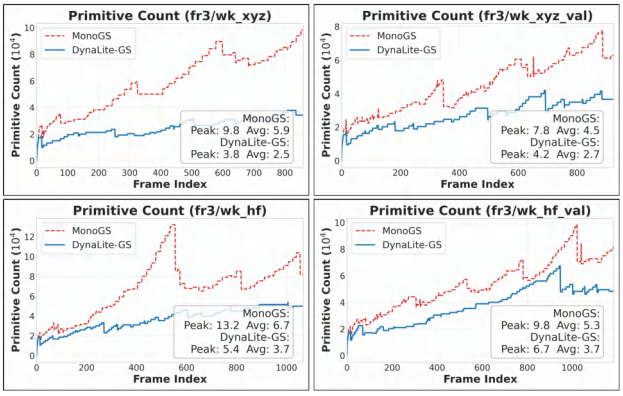

Fig. 6. Temporal Evolution of Primitive Count on Long Sequences from the TUM Dataset.

In contrast, the primitive count (Fig. 6) and cumulative computational load (Fig. 7) curves show that DynaLite-GS consistently maintains exceptional map compactness and a flat load growth. Quantitatively, our method reduces the peak primitive count by up to 61.2% and decreases the cumulative computational load (i.e., the Area Under the Curve of the primitive count) by up to 57.8%.

This lightweight advantage stems from an efficient architecture: the front-end DMS isolates dynamic pixels with low cost, blocking invalid primitive generation; the back-end STBM corrects tracking deviations caused by dynamic occlusions, preventing compensatory map divergence induced by pose drift. This streamlined map scale fundamentally minimizes per-frame sorting and rasterization, thereby reducing the overall computational load.

Benefiting from this compactness, DynaLite-GS filters dynamics without heavy optical flow or segmentation networks. Compared to state-of-the-art methods (Table V), it reduces computational demands and peak GPU memory by up to 83.3%, enabling high-quality consumer-grade deployment.

V. Conclusion

We present DynaLite-GS, a lightweight dynamic 3D Gaussian Splatting SLAM system tailored for resource-constrained platforms. To address the computational bottlenecks of existing methods, we propose a Decoupled Masking Strategy that efficiently filters dynamic disturbances by integrating coarse semantics, geometric depth consistency, and visibility in a coarse-to-fine manner. Additionally, a Stochastic Backtracking Mechanism tailored for dynamic scenes mitigates drift and ensures global consistency by randomly optimizing historical keyframes. Experiments demonstrate

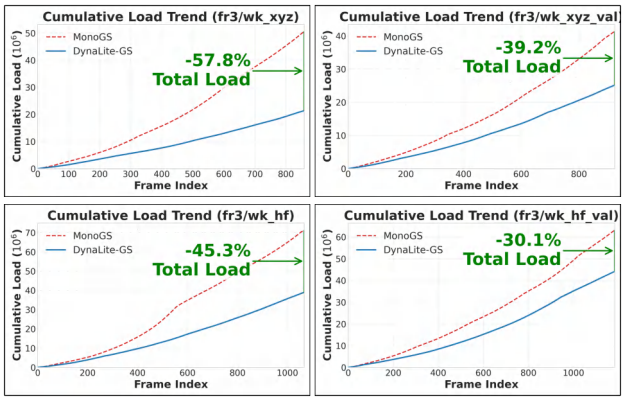

Fig. 7. Cumulative Computational Load Analysis.

that our system achieves tracking accuracy superior to state-of-the-art methods with competitive mapping quality, all while operating with high efficiency and low memory consumption. However, the current approach prioritizes filtering dynamic regions over explicitly modeling them, which may limit robustness in scenes where dynamic objects dominate the field of view. Additionally, the evaluation is restricted to indoor benchmarks, and deployment on strictly embedded platforms remains to be validated. Future work will focus on large-scale outdoor environments, handling extreme motion blur, and further engineering optimizations for real-time embedded performance.

REFERENCES

[1] 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*, vol. 65, no. 1, pp. 99–106, 2021.

[2] 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*, 2022, pp. 12 786–12 796.

[3] H. Jiang, Y. Xu, K. Li, J. Feng, and L. Zhang, "Rodyn-slam: Robust dynamic dense rgb-d slam with neural radiance fields," *IEEE Robotics and Automation Letters*, 2024.

[4] 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*, 2023, pp. 17 408–17 419.

[5] 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*, 2023, pp. 13 293–13 302.

[6] B. Fei, J. Xu, R. Zhang, Q. Zhou, W. Yang, and Y. He, "3d gaussian splatting as new era: A survey," *IEEE Transactions on Visualization and Computer Graphics*, 2024.

[7] 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*, 2024, pp. 18 039–18 048.

[8] C. Yan, D. Qu, D. Xu, B. Zhao, Z. Wang, D. Wang, and X. Li, "Gs-slam: Dense visual slam with 3d gaussian splatting," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 19 595–19 604.

[9] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[10] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.

[11] G. Jocher, A. Chaurasia, and J. Qiu, "Ultralytics yolo," https://github.com/ultralytics/ultralytics, 2024.

[12] B. Bescos, J. M. Fácil, J. Civera, and J. Neira, "Dynaslam: Tracking, mapping, and inpainting in dynamic scenes," *IEEE robotics and automation letters*, vol. 3, no. 4, pp. 4076–4083, 2018.

[13] C. Yu, Z. Liu, X.-J. Liu, F. Xie, Y. Yang, Q. Wei, and Q. Fei, "Ds-slam: A semantic visual slam towards dynamic environments," in *2018 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2018, pp. 1168–1174.

[14] Z. Xu, J. Niu, Q. Li, T. Ren, and C. Chen, "Nid-slam: Neural implicit representation-based rgb-d slam in dynamic environments," in *2024 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2024, pp. 1–6.

[15] M. Li, Z. Guo, T. Deng, Y. Zhou, Y. Ren, and H. Wang, "Ddn-slam: Real time dense dynamic neural implicit slam," *IEEE Robotics and Automation Letters*, 2025.

[16] Y. Xu, H. Jiang, Z. Xiao, J. Feng, and L. Zhang, "Dg-slam: Robust dynamic gaussian splatting slam with hybrid pose optimization," *Advances in Neural Information Processing Systems*, vol. 37, pp. 51 577–51 596, 2024.

[17] M. Li, Y. Zhou, H. Zhou, X. Hu, F. Roemer, H. Wang, and A. Osman, "Dy3dgs-slam: Monocular 3d gaussian splatting slam for dynamic environments," in *2025 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2025, pp. 14 572–14 578.

[18] J. Zheng, Z. Zhu, V. Bieri, M. Pollefeys, S. Peng, and I. Armeni, "Wildgs-slam: Monocular gaussian splatting slam in dynamic environments," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 11 461–11 471.

[19] W. Zheng, L. Ou, J. He, L. Zhou, X. Yu, and Y. Wei, "Up-slam: Adaptively structured gaussian slam with uncertainty prediction in dynamic environments," *arXiv preprint arXiv:2505.22335*, 2025.

[20] Z. Teed and J. Deng, "Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras," *Advances in neural information processing systems*, vol. 34, pp. 16 558–16 569, 2021.

[21] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby *et al.*, "Dinov2: Learning robust visual features without supervision," *arXiv preprint arXiv:2304.07193*, 2023.

[22] M. Kong, J. Lee, S. Lee, and E. Kim, "Dgs-slam: Gaussian splatting slam in dynamic environment," *arXiv preprint arXiv:2411.10722*, 2024.

[23] L. Wen, S. Li, Y. Zhang, Y. Huang, J. Lin, F. Pan, Z. Bing, and A. Knoll, "Gassidy: Gaussian splatting slam in dynamic environments," in *2025 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2025, pp. 8471–8477.

[24] 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*, 2024, pp. 21 357–21 366.

[25] E. Sandström, G. Zhang, K. Tateno, M. Oechsle, M. Niemeyer, Y. Zhang, M. Patel, L. Van Gool, M. Oswald, and F. Tombari, "Splat-slam: Globally optimized rgb-only slam with 3d gaussians," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 1680–1691.

[26] L. Li, L. Zhang, Z. Wang, and Y. Shen, "Gs3lam: Gaussian semantic splatting slam," in *Proceedings of the 32nd ACM International Conference on Multimedia*, 2024, pp. 3019–3027.

[27] C. Huang, L. Zhang, T. Deng, K. Wang, and M. Li, "Mpdg-slam: Motion probability-based 3dgs-slam in dynamic environment," in *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2025, pp. 14 045–14 052.

[28] 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*. IEEE, 2012, pp. 573–580.

[29] E. Palazzolo, J. Behley, P. Lottes, P. Giguere, and C. Stachniss, "Refusion: 3d reconstruction in dynamic environments for rgb-d cameras exploiting residuals," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 7855–7862.

[30] 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*, vol. 37, no. 6, pp. 1874–1890, 2021.