DG-SLAM: Robust Dynamic Gaussian Splatting SLAM with Hybrid Pose Optimization

Yueming Xu[1*] **Haochen Jiang**[1*] **Zhongyang Xiao**[2] **Jianfeng Feng**[1] **Li Zhang**[1†]

[1]Fudan University [2]Autonomous Driving Division, NIO

https://github.com/fudan-zvg/DG-SLAM

Abstract

Achieving robust and precise pose estimation in dynamic scenes is a significant research challenge in Visual Simultaneous Localization and Mapping (SLAM). Recent advancements integrating Gaussian Splatting into SLAM systems have proven effective in creating high-quality renderings using explicit 3D Gaussian models, significantly improving environmental reconstruction fidelity. However, these approaches depend on a static environment assumption and face challenges in dynamic environments due to inconsistent observations of geometry and photometry. To address this problem, we propose DG-SLAM, the first robust dynamic visual SLAM system grounded in 3D Gaussians, which provides precise camera pose estimation alongside high-fidelity reconstructions. Specifically, we propose effective strategies, including motion mask generation, adaptive Gaussian point management, and a hybrid camera tracking algorithm to improve the accuracy and robustness of pose estimation. Extensive experiments demonstrate that DG-SLAM delivers state-of-the-art performance in camera pose estimation, map reconstruction, and novel-view synthesis in dynamic scenes, outperforming existing methods meanwhile preserving real-time rendering ability.

1 Introduction

Visual simultaneous localization and mapping (SLAM), the task of reconstructing a 3D map within an unknown environment while simultaneously estimating the camera pose, is recognized as a critical component to achieving autonomous navigation in novel 3D environments for mobile robots [1]. It has been widely used in various forms in fields such as robotics, autonomous driving, and augmented/virtual reality (AR/VR). However, the majority of previous research [2, 3, 4, 5, 6, 7, 8, 9, 10] typically relies on the assumption of static environments, limiting the practical applicability of this technology in daily life. Consequently, how to achieve accurate and robust pose estimation in dynamic scenes remains an urgent problem to be addressed for mobile robots.

Recently, many researchers [6, 7, 8, 9, 10] have endeavored to replace the conventional explicit representations used in visual SLAM, such as Signed Distance Function (SDF), voxel grids [11], meshes [12], and surfel clouds [13], with the neural radiance field (NeRF) [14] approach for reconstructing the neural implicit map. This novel map representation is more continuous, efficient, and able to be optimized with differentiable rendering, which has the potential to benefit applications like navigation and reconstruction. However, these methods exhibit two primary issues: the scene's bounds are required to be predefined to initialize the neural voxel grid; and the implicit representation proves challenging for information fusion and editing. To address these problems, recent works

*Yueming Xu and Haochen Jiang contribute equally to this work.

†Li Zhang (lizhangfd@fudan.edu.cn) is the corresponding author with School of Data Science, Fudan University.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

 https://doi.org/10.52202/079017-1633

like GS-SLAM [15], SplaTam [16], and Gaussian splatting SLAM [17] leverage the 3D-GS [18] to explicit represent the scene's map. This explicit geometric representation is also smooth, continuous, and differentiable. Moreover, a substantial array of Gaussians can be rendered with high efficiency through splatting rasterization techniques, achieving up to 300 frames per second (FPS) at a resolution of 1080p. However, all these above-mentioned neural SLAM methods do not perform well in dynamic scenes. The robustness of these systems significantly decreases, even leading to tracking failures, when dynamic objects appear in the environment.

To tackle these problems, we propose a novel 3D Gaussian-based visual SLAM that can reliably track camera motion in dynamic indoor environments. Due to the capability of 3D-GS to accomplish high-quality rendering in real-time, the SLAM system more readily converges to a global optimum during pose optimization, thereby achieving better and more stable pose optimization results. A cornerstone of our approach to achieving robust pose estimation lies in the innovative motion mask generation algorithm. This algorithm filters out sampled pixels situated within invalid zones, thereby refining the estimation process. In addition to the constraint of depth residual, we employ a spatio-temporal consistency strategy within an observation window to generate depth warp masks. By incrementally fusing the depth warp mask and semantic mask, the motion mask will become more precise to reflect the true motion state of objects. To improve the accuracy and stability of pose estimation, we leverage DROID-SLAM [19] odometry (DROID-VO) to provide an initial pose estimate and devise a coarse-to-fine optimization algorithm built upon the initially estimated camera pose. This aims to minimize the disparity between pose estimation and the reconstructed map, employing photorealistic alignment optimization through Gaussian Splatting. Moreover, this hybrid pose optimization approach effectively ensures the accuracy and quality of the generated depth warp mask, thereby facilitating better performance in the next camera tracking stage. To obtain high-quality rendering results, we propose a novel adaptive Gaussian point addition and pruning method to keep the geometry clean and enable accurate and robust camera tracking. Capitalizing on the factor graph structure inherent in DROID-SLAM, our system is capable of executing dense Bundle Adjustment (DBA) upon completion of tracking to eliminate accumulated errors.

In summary, our **contributions** are summarized as follows: **(i)** To the best of our knowledge, this is the first robust dynamic Gaussian splatting SLAM with hybrid pose optimization, capable of achieving both real-time rendering and high-fidelity reconstruction performance. **(ii)** To mitigate the impact of dynamic objects during pose estimation, we propose an advanced motion mask generation strategy that integrates spatio-temporal consistent depth masks with semantic priors, thereby significantly enhancing the precision of motion object segmentation. **(iii)** We design a hybrid camera tracking strategy utilizing the coarse-to-fine pose optimization algorithm to improve the consistency and accuracy between the estimated pose and the reconstructed map. **(iv)** To better manage and expand the Gaussian map, we propose an adaptive Gaussian point addition and pruning strategy. It ensures geometric integrity and facilitates accurate camera tracking. **(v)** Extensive evaluations on two challenging dynamic datasets and one common static dataset demonstrate the state-of-the-art performance of our proposed SLAM system, particularly in real-world scenarios.

2 Related work

Visual SLAM with dynamic objects filter. Dynamic object filtering is pivotal for reconstructing static scenes and bolstering the robustness of pose estimation. Existing approaches fall into two main categories: the first relies on re-sampling and residual optimization strategies to remove outliers, as seen in works such as ORB-SLAM2 [2], ORB-SLAM3 [3], and Refusion [5]. These methods, however, are generally limited to addressing small-scale motions and often falter in the face of extensive, continuous object movements. The second group employs the additional prior knowledge, for example, semantic segmentation or object detection prior [20, 21, 22, 23, 24, 25, 26], to remove the dynamic objects. However, all these methods often exhibit a domain gap when applied in real-world environments, leading to the introduction of prediction errors. More recently, deep neural networks have been employed to build end-to-end visual odometry, which performs better in specific environments such as DROID-SLAM [19], DytanVO [27], and DeFlowSLAM [28]. However, these methods still require a substantial amount of training data and are unable to reconstruct the high-fidelity static map.

RGB-D SLAM with neural implicit representation. Neural implicit scene representations, also known as neural fields [29], have attracted considerable attention in the field of RGB-D SLAM for

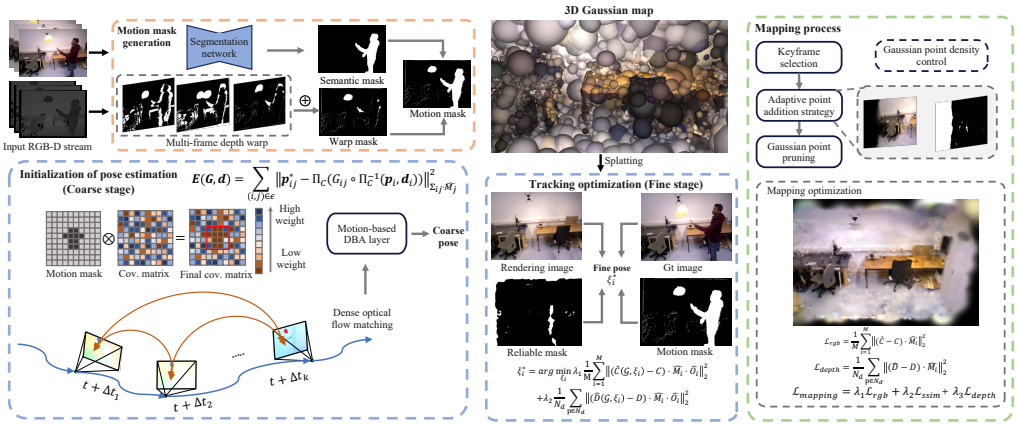

Figure 1: **Overview of DG-SLAM.** Given a series of RGB-D frames, we reconstruct the static high-fidelity 3D Gaussian map and optimize the camera pose represented with lie algebra ξ_i.

their impressive expressiveness and low memory footprint. Initial studies, including iMap [6] and DI-Fusion [30], explored the utilization of a single MLP and a feature grid to encode scene geometries within a latent space. However, they both share a critical issue: the problem of network forgetting, which is catastrophic for long-term localization and mapping. In response to this limitation, NICE-SLAM [7] introduces a hierarchical feature grid approach to enhance scene representation fidelity and implements a localized feature updating strategy to address the issue of network forgetting. While these advancements contribute to improved accuracy, they necessitate greater memory consumption and may impact the system's ability to operate in real-time. More recently, existing methods like Vox-Fusion [10], Co-SLAM [8], and ESLAM [9] explore sparse encoding or tri-plane representation strategy to improve the quality of scene reconstruction and the system's execution efficiency. Point-SLAM [31] draws inspiration from the concept of Point-NeRF [32], utilizing neural points to encode spatial geometry and color features. It employs an explicit method to represent spatial maps, effectively improving the accuracy of localization and mapping. All these methods have demonstrated impressive results based on the strong assumptions of static scene conditions. The robustness of these systems significantly decreases when dynamic objects appear in the environment. Recently, Rodyn-SLAM [33] proposed utilizing optical flow and semantic segmentation prior to filter out dynamic objects, and employing a neural rendering method as the frontend. However, this approach is computationally intensive and limits the maximum accuracy achievable in pose estimation.

3D Gaussian splatting SLAM methods. Compared to the aforementioned NeRF-based SLAM methods, 3D Gaussian splatting (3D-GS) [18] has garnered widespread interest among researchers due to its advantages in high-fidelity and real-time rendering. Unlike previous implicit map representation methods, 3D-GS explicitly models scene maps by independent Gaussian spheres, naturally endowing it with geometric structure property. Some researchers [17, 15, 34, 16, 35] are exploring the replacement of implicit representations (NeRF) with 3D-GS in the mapping thread. However, these methods are currently constrained by the assumption of static environments, rendering them ineffective in dynamic scenes. It significantly restricts the practical application of Gaussian SLAM systems in real-life scenarios. Under the premise of ensuring high-fidelity reconstruction and real-time rendering, our system is designed to improve the accuracy and robustness of pose estimation under dynamic environments.

3 Approach

Given a sequence of RGB-D frames $\{I_i, D_i\}_{i=1}^{N}, I_t \in \mathbb{R}^3, D_t \in \mathbb{R}$, our method (Fig. 1) aims to simultaneously recover camera poses $\{\xi_i\}_{i=1}^{N}, \xi_t \in \mathbb{SE}(3)$ and reconstruct the static 3D scene map represented by 3D Gaussian sphere in dynamic environments. Similar to most modern SLAM systems [36, 37], our system comprises two distinct processes: the tracking process as the frontend and the mapping process as the backend.

https://doi.org/10.52202/079017-1633

3.1 3D Gaussian map representation

To obtain real-time rendering and high-fidelity reconstruction mapping, we represent the scene as a set of 3d Gaussian ellipsoid $\mathcal{G}$, which simultaneously possesses geometric and appearance properties.

$$\mathcal{G} = \{\mathcal{G}_i : (\mu_i, \mathbf{\Sigma}_i, \alpha_i, \mathbf{h}_i) \big| \forall \mathcal{G}_i \in \mathcal{G}\} \tag{1}$$

Each 3D Gaussian ellipsoid $\mathcal{G}_i$ is composed of its center position $\mu_i \in \mathbb{R}^3$, covariance matrix $\mathbf{\Sigma}_i \in \mathbb{R}^{3\times3}$, opacity $\alpha_i \in \mathbb{R}$, and spherical harmonics coefficients $\mathbf{h}_i \in \mathbb{R}^{16}$.

Color and depth splatting rendering. To obtain the rendering image of color and depth, we project the 3D Gaussian $(\mu_w, \mathbf{\Sigma}_w)$ in world coordinate to 2D Gaussian $(\mu_I, \mathbf{\Sigma}_I)$ on the image plane:

$$\mu_I = \pi\left(\boldsymbol{T}_w^c \cdot \mu_w\right), \quad \mathbf{\Sigma}_I = \mathbf{J}\mathbf{R}\mathbf{\Sigma}_w\mathbf{R}^T\mathbf{J}^T, \tag{2}$$

where $\mathbf{R}$ is the viewing transformation and $\mathbf{J}$ is the Jacobian of the affine approximation of the projective transformation. Following the alpha blending method in 3DGS [18], we accumulate the splatting Gaussian ellipsoid along the observation image pixel at the current estimation pose ξ_i to render the color and depth as:

$$\hat{C} = \sum_{i\in\mathcal{G}} \mathbf{c}_i f_i(\mu_I, \mathbf{\Sigma}_I) \prod_{j=1}^{i-1} \left(1 - f_j(\mu_I, \mathbf{\Sigma}_I)\right), \quad \hat{D} = \sum_{i\in\mathcal{G}} \mathbf{d}_i f_i(\mu_I, \mathbf{\Sigma}_I) \prod_{j=1}^{i-1} \left(1 - f_j(\mu_I, \mathbf{\Sigma}_I)\right), \tag{3}$$

where $f_i(\cdot)$ is the Gaussian distribution function weighted by opacity α_i. $\mathbf{c}_i$ represents the color of the projected Gaussian point computed by learnable spherical harmonics coefficients $\mathbf{h}_i$. Similarly, $\mathbf{d}_i$ denotes the depth of Gaussian point $\mathcal{G}_i$, obtained by projecting to z-axis in the camera coordinate.

Accumulated opacity. We use accumulated opacity $\hat{O}$ to represent the rendering reliability for each pixel and judge whether the Gaussian map is well-optimized.

$$\hat{O} = \sum_{i\in\mathcal{G}} f_i(\mu_I, \mathbf{\Sigma}_I) \prod_{j=1}^{i-1} \left(1 - f_j(\mu_I, \mathbf{\Sigma}_I)\right), \tag{4}$$

3.2 Motion mask generation

For each input keyframe, we select its associated keyframe set $\mathcal{D}$ within a sliding window. To reduce the computation and improve the accuracy of generating motion mask, we employ the depth warping operation solely on keyframes. To ensure the overlap between adjacent keyframes is not too small, we employ optical-flow distance to determine keyframe insertion. In regions with more intense motion, our goal is to insert as many keyframes as possible.

For the pixel p in keyframe i, we reproject it onto keyframe j as follows:

$$\boldsymbol{p}_{i\to j} = f_{warp}\left(\xi_{ji}, \boldsymbol{p}_i, D_i(\boldsymbol{p}_i)\right) = \boldsymbol{K}\boldsymbol{T}_{ji}\left(\boldsymbol{K}^{-1}D_i(\boldsymbol{p}_i)\boldsymbol{p}_i^{homo}\right), \tag{5}$$

where $\boldsymbol{K}$ and $\boldsymbol{T}_{ji}$ represent the intrinsic matrix and the transformation matrix between frame i and frame j, respectively. $\boldsymbol{p}_i^{homo} = (u, v, 1)$ is the homogeneous coordinate of $\boldsymbol{p}_i$.

Given any associate keyframes $D_i, D_j \in \mathcal{D}$, we utilize warp function f_{warp} to compute the residual of reprojection depth value. By setting a suitable threshold e_{th}, we derive the depth warp mask $\widehat{\mathcal{M}}_{j,i}^{wd}$ corresponding to dynamic objects as:

$$\widehat{\mathcal{M}}_{j,i}^{wd} : \left\{\bigcap_{\boldsymbol{p}_i\in D_i} \mathbf{1}(D_j(\boldsymbol{p}_{i\to j}) - D_i(\boldsymbol{p}_i) < e_{th}) \otimes \boldsymbol{I}_{m\times n}\right\} \tag{6}$$

where $\boldsymbol{I}_{m\times n}$ represents a matrix of the same size as the image, filled with ones. The $\otimes$ operation signifies that for each element in the matrix $\boldsymbol{I}_{m\times n}$, we assess whether its warp depth meets a specified threshold and subsequently modify the corresponding value at that position. As illustrated in Fig. 1, to derive a more precise warp mask, we consider the spatial coherence of object motion within a sliding window of length N and combine the multiple observation warp masks. Unlike ReFusion [5], our method can mitigate the potential impact of observation noise from a single warp mask. When object motion becomes significant, we only mask the foreground pixels where the depth residual is positive to avoid a large portion of pixels being masked as dynamic regions. Subsequently, we integrate the warp mask and semantic mask to derive the final motion mask $\widehat{\mathcal{M}}_j$ as:

$$\widehat{\mathcal{M}}_j = \widehat{\mathcal{M}}_{j,i}^{wd} \cap \widehat{\mathcal{M}}_{j,i-1}^{wd} \cap \widehat{\mathcal{M}}_{j,i-2}^{wd} \cdots \cap \widehat{\mathcal{M}}_{j,i-N}^{wd} \cup \widehat{\mathcal{M}}_j^{sg}, \tag{7}$$

Thanks to our innovative approach to generating motion masks, the omission of dynamic objects by semantic priors can be effectively compensated. Furthermore, by leveraging a spatial consistency strategy, the inaccuracy of edge region recognition during depth warping can be significantly reduced.

3.3 Coarse-to-fine camera tracking.

The constant speed motion model struggles to infer a reasonable initial optimized pose value in dynamic scenes. An inaccurate initial pose will lead to optimization getting trapped in local optima more easily and can affect the generation quality of the depth warp mask. To achieve more precise pose estimation in dynamic environments, we utilize the visual odometry (VO) component from DROID-SLAM [19] as the coarse pose estimation results in camera tracking. We also conduct a dense bundle adjustment in every interaction for a set of keyframes to optimize the corresponding pose $\mathbf{G}$ and depth $\mathbf{d}$:

$$\mathbf{E}(\mathbf{G}, \mathbf{d}) = \sum_{(i,j)\in\epsilon} \left\| \mathbf{p}_{ij}^{*} - \Pi_C \left(G_{ij} \circ \Pi_C^{-1} \left(\mathbf{p}_i, \mathbf{d}_i \right) \right) \right\|_{\Sigma_{ij}\cdot\widehat{\mathcal{M}}_j}^{2}, \tag{8}$$

where $\Sigma_{ij} = \operatorname{diag}(w_{ij})$, w_{ij} represents the confidence weights. G_{ij} denotes the relative transformation between the poses G_i and G_j. $\mathbf{p}_i$ means a grid of pixel coordinates. Moreover, $\mathbf{p}_{ij}^{*}$ is corrected correspondence as predicted by updated optical flow estimation. To overcome the influence of dynamic objects on bundle adjustment, we introduce suppression through the motion mask $\widehat{\mathcal{M}}_j$ applied to the weighted covariance matrix. Consequently, the weighted confidence associated with dynamic objects is reduced to zero.

To further improve the accuracy of pose estimation and minimize inconsistencies between the estimated pose and the reconstructed map, we implement fine camera tracking by leveraging Gaussian splatting, based on the initial obtained pose. Due to obstructions by dynamic objects and the inadequate optimization of Gaussian points in the map, the rendered image may exhibit blurring or black floaters. Therefore, we employ the accumulated opacity, as outlined in 4, to denote the pixel rendering reliability. The reliable mask $\widehat{\mathcal{O}}_i$ for camera tracking is generated as follows:

$$\widehat{\mathcal{O}}_i : \left\{ \bigcap_{[\mu_j, v_j]\in I_i} \mathbf{1}(\hat{O}[\mu_j, v_j] > \tau_{track}) \otimes \boldsymbol{I}_{m\times n} \right\}, \tag{9}$$

where μ_j, v_j represents the pixel location. When the accumulated opacity at the pixel exceeds a given threshold τ_{track}, we consider the Gaussian point associated with that pixel to be well-optimized. Consequently, using the rendered image at these pixels for pose estimation is deemed reliable. The overall loss function is finally formulated as the following minimization:

$$\xi_i^{*} = \arg\min_{\xi_i} \lambda_1 \frac{1}{M} \sum_{i=1}^{M} \left\| \left(\hat{C}(\mathcal{G}, \xi_i) - C \right) \cdot \widehat{\mathcal{M}}_i \cdot \widehat{\mathcal{O}}_i \right\|_2^2 + \lambda_2 \frac{1}{N_d} \sum_{\mathbf{p}\in N_d} \left\| \left(\hat{D}(\mathcal{G}, \xi_i) - D \right) \cdot \widehat{\mathcal{M}}_i \cdot \widehat{\mathcal{O}}_i \right\|_2^2. \tag{10}$$

Where ξ_i denotes the camera pose requiring optimization. C and D denote the ground truth color and depth map, respectively. M represents the number of sampled pixels in the current image. Note that only pixels with valid depth value N_d are considered in optimization.

This hybrid pose optimization approach effectively ensures the accuracy and quality of the warp mask, thereby facilitating better performance in the next camera tracking stage. In short, this hybrid pose optimization strategy enables us to achieve more precise and robust pose estimation whether in dynamic or static environments.

3.4 SLAM system

Map initialization. For the first frame, we do not conduct the tracking step and the camera pose is set to the identity matrix. To better initialization, we reduce the gradient-based dynamic radius to half so that can add more Gaussian points. For pixels located outside the motion mask, we sample and reproject them to the world coordinates. We initialize the Gaussian point color and center position μ_i with the RGB value and reprojection coordinate of the pixel, respectively. The opacity α_i is set as 0.1 and the scale vector $\mathbf{S}_i$ is initialized based on the mean distance of the three nearest neighbor points. The rotation matrix $\mathbf{R}_i$ is initialized as the identity matrix.

Map optimization. To optimize the scene Gaussian representation, we render depth and color in independent view as seen Eq. 3, comparing with the ground truth map:

$$\mathcal{L}_{rgb} = \frac{1}{M} \sum_{i=1}^{M} \left\| \left(\hat{C} - C \right) \cdot \widehat{\mathcal{M}}_i \right\|_2^2, \quad \mathcal{L}_{depth} = \frac{1}{N_d} \sum_{\mathbf{p}\in N_d} \left\| \left(\hat{D} - D \right) \cdot \widehat{\mathcal{M}}_i \right\|_2^2, \tag{11}$$

In contrast to existing methods, we introduce the motion mask $\widehat{\mathcal{M}}_i$ to remove sampled pixels within the dynamic region effectively. The final Gaussian map optimization is performed by minimizing the

https://doi.org/10.52202/079017-1633

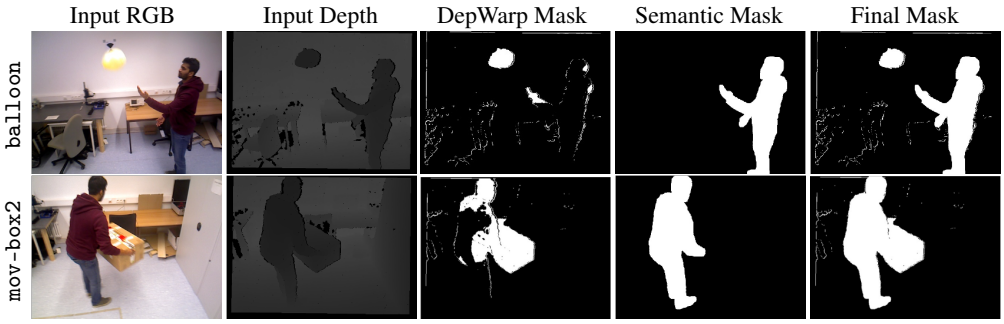

Figure 2: **Qualitative results of the motion mask generation.** By fusing the semantic mask and depth warp mask, the final mask will be more precise.

geometric depth loss and photometric color loss :

$$\mathcal{L}_{mapping} = \lambda_1 \mathcal{L}_{rgb} + \lambda_2 \mathcal{L}_{ssim} + \lambda_3 \mathcal{L}_{depth}, \tag{12}$$

where $\mathcal{L}_{ssim}$ denotes the structural similarity loss between two images. Moreover, $\lambda_1, \lambda_2, \lambda_3$ are weight factors for balance in the optimization process.

Adapative Gaussian point adding strategy. To guarantee the superior scene representation capability of 3D Gaussian, we also employ dynamic Gaussian point density, which is inspired by Point-SLAM [31]. The insertion radius is determined based on color gradient, allowing for a denser allocation of points in areas with high texture while minimizing point addition in regions of low texture. This method efficiently manages memory consumption by adapting the point density according to the textural complexity of the scene.

To ensure the points we add are both necessary and effective, we adopt a two-stage adaptive point-add strategy. For new viewpoints without previous observation, we initially perform uniform sampling across the entire image to ensure that new observed areas can be covered by Gaussian points. Moreover, if the accumulated opacity $\hat{O}$, calculated by Eq. 4, falls below the threshold o_{th}, or the depth residual between the rendered pixels and the ground truth depth is excessively large, we then add 3D Gaussian points based on these under-fitting pixels. At last, these new Gaussian points will be initialized based on the point density.

Map point deleting strategy. Given that the added 3D Gaussian points have not been subjected to geometric consistency verification and may exhibit unreasonable representation values during optimization, this could lead to the generation of a low-quality dense map or the introduction of numerous artifacts in the rendering image. we implement the pruning operation as part of the Gaussian map optimization. To ensure the points we delete are both reasonable and accurate, we also adopt a two-stage point delete strategy. For all Gaussian points observed from the current viewpoint, we delete points based on three criteria: the opacity value, the maximum scale, and the ratio between the ellipsoid's major and minor axes, described as follow:

$$\alpha_i < \tau_\alpha \quad \text{or} \quad \max(\mathbf{S}) > \tau_{s1} \quad \text{or} \quad \frac{\max(\mathbf{S})}{\min(\mathbf{S})} > \tau_{s2}. \tag{13}$$

Moreover, due to potential inaccuracies along the edges of the current motion mask, adding these points could result in artifacts within the scene map. Thus, we project these points on keyframes in a small sliding window to recheck whether they can be observed by these keyframes. If the number of observations of a point is too low, we consider its addition to be insufficiently robust, and therefore, it will be removed from the current Gaussian map.

4 Experiments

Datasets. Our methodology is evaluated using three publicly available challenging datasets: *TUM RGB-D* dataset [38], *BONN RGB-D Dynamic* dataset [5] and *ScanNet* [39]. These three datasets contain both challenging dynamic scenes and real static scenes. This selection facilitates a comprehensive assessment of our approach under varied conditions, demonstrating its applicability and robustness in real-world indoor environments.

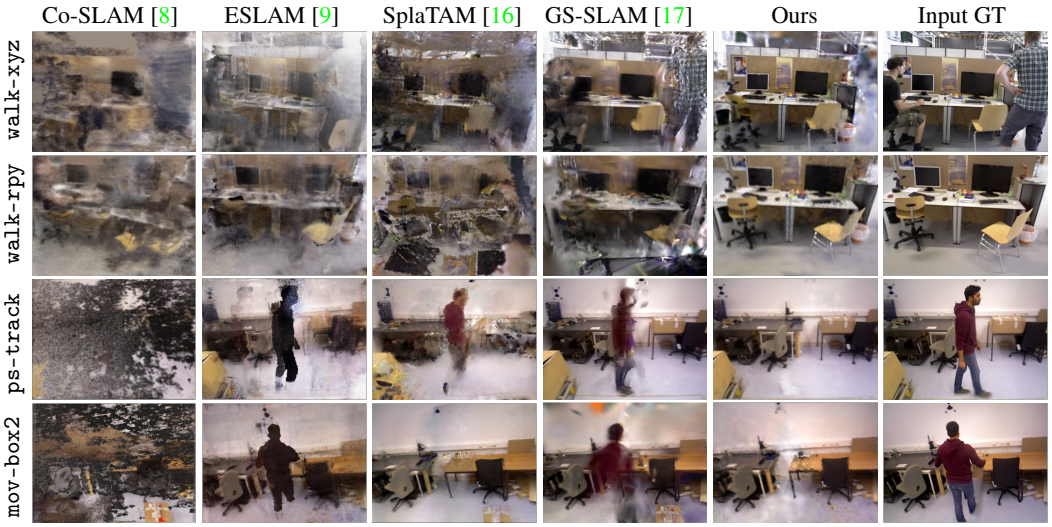

Figure 3: **Visual comparison of the rendering image on the *TUM* and *BONN* datasets.** Our results are more complete and accurate without the dynamic object floaters.

		ball	ball2	ps_trk	ps_trk2	mv_box2	Avg.
NICE-SLAM[7]	**Acc.**[cm]↓	X	24.30	43.11	74.92	17.56	39.97
	Comp.[cm]↓	X	16.65	117.95	172.20	18.19	81.25
	Comp. Ratio[≤ 5cm%]↑	X	29.68	15.89	13.96	32.18	22.93
Co-SLAM[8]	**Acc.**[cm]↓	10.61	14.49	26.46	26.00	12.73	18.06
	Comp.[cm]↓	10.65	40.23	124.86	118.35	10.22	60.86
	Comp. Ratio[≤ 5cm%]↑	34.10	3.21	2.05	2.90	39.10	16.27
ESLAM[9]	**Acc.**[cm]↓	17.17	26.82	59.18	89.22	12.32	40.94
	Comp.[cm]↓	**9.11**	13.58	145.78	186.65	10.03	73.03
	Comp. Ratio[≤ 5cm%]↑	47.44	47.94	20.53	17.33	41.41	34.93
DG-SLAM(Ours)	**Acc.**[cm]↓	**7.00**	**5.80**	**9.14**	**11.78**	**6.56**	**8.06**
	Comp.[cm]↓	9.80	**8.05**	**17.99**	**20.10**	**7.61**	**15.46**
	Comp. Ratio[≤ 5cm%]↑	**49.46**	**52.41**	**34.62**	**32.81**	**49.02**	**43.67**

Table 1: **Reconstruction results on several dynamic scene sequences in the *BONN* dataset**. Instances of tracking failures are denoted by "X". The most superior outcomes within the domain of RGB-D SLAMs are highlighted in **bold** for emphasis.

Metrics. For the evaluation of pose estimation, we utilize the Root Mean Square Error (RMSE) and Standard Deviation (STD) of Absolute Trajectory Error (ATE) [38]. Prior to assessment, the estimated trajectory is aligned with the ground truth trajectory via Horn's Procrustes method. [40], ensuring a coherent basis for evaluation. To evaluate the reconstruction quality of static maps in dynamic scenes, we employ three metrics(i) *Accuracy* (cm), (ii) *Completion* (cm), and (iii) *Completion Ratio* (percentage of points within a 5cm threshold), following NICE-SLAM [7]. Since the BONN dataset provides only the ground truth point cloud, we randomly sampled 200,000 points from the GT point cloud and the reconstructed mesh surface to calculate these metrics.

Implementation details. We run our DG-SLAM on an RTX 3090 Ti GPU at 2 FPS on BONN datasets, which takes roughly 9GB of memory. We set the loss weight $\lambda_1 = 0.9$, $\lambda_2 = 0.2$ and $\lambda_3 = 0.1$ to train our model. The number of iterations for the tracking and mapping processes has been set to 20 and 40, respectively. For the Gaussian points deleting, we set $\tau_\alpha = 0.005$, $\tau_{S1} = 0.4$ and $\tau_{S2} = 36$ to avoid the generation of abnormal Gaussian points. What's more, we utilize Oneformer [41] to generate prior semantic segmentation. For the depth wrap mask, we set the window size to 4 and the depth threshold to 0.6. We also adopt the keyframe selection strategy from DROID-VO [19] based on optical flow.

https://doi.org/10.52202/079017-1633

Method	f3/w_r	f3/w_x	f3/w_s	f3/s_x	f2/d_p	f3/l_o	Avg.
ORB-SLAM3 [3]	68.7	28.1	2.0	**1.0**	**1.5**	**1.0**	17.1
ReFusion [5]	-	9.9	1.7	4.0	-	-	5.2
Co-fusion [42]	-	69.6	55.1	2.7	-	-	42.5
MID-fusion [43]	-	6.8	2.3	6.2	-	-	5.1
EM-fusion [44]	-	6.6	1.4	3.7	-	-	3.9
iMAP*[6]	139.5	111.5	137.3	23.6	119.0	5.8	89.5
NICE-SLAM[7]	X	113.8	88.2	7.9	X	6.9	54.2
Vox-Fusion[10]	X	146.6	109.9	3.8	X	26.1	71.6
Co-SLAM[8]	52.1	51.8	49.5	6.0	7.6	2.4	28.3
ESLAM[9]	90.4	45.7	93.6	7.6	X	2.5	48.0
Rodyn-SLAM[33]	7.8	8.3	1.7	5.1	5.6	2.8	5.3
SplaTAM[16]	100.4	218.3	115.2	1.7	5.4	5.1	74.4
GS-SLAM[17]	33.5	37.7	8.4	2.7	8.6	1.8	15.5
DROID-VO[19]	10.0	1.7	0.7	1.1	3.7	2.3	3.3
DG-SLAM(Ours)	**4.3**	**1.6**	**0.6**	**1.0**	3.2	2.3	**2.2**

Table 2: **Camera tracking results on several dynamic scene sequences in the *TUM* dataset**. "$*$" denotes the version reproduced by NICE-SLAM. "X" and "-" denote the tracking failures and absence of mention, respectively. The metric is Absolute Trajectory Error (ATE) and the unit is [cm].

Method	ball	ball2	ps_tk	ps_tk2	ball_tk	mv_box2	Avg.
ORB-SLAM3 [3]	5.8	17.7	70.7	77.9	**3.1**	**3.5**	29.8
ReFusion [5]	17.5	25.4	28.9	46.3	30.2	17.9	27.7
iMAP*[6]	14.9	67.0	28.3	52.8	24.8	28.3	36.1
NICE-SLAM[7]	X	66.8	54.9	45.3	21.2	31.9	44.1
Vox-Fusion[10]	65.7	82.1	128.6	162.2	43.9	47.5	88.4
Co-SLAM[8]	28.8	20.6	61.0	59.1	38.3	70.0	46.3
ESLAM[9]	22.6	36.2	48.0	51.4	12.4	17.7	31.4
Rodyn-SLAM[33]	7.9	11.5	14.5	13.8	13.3	12.6	12.3
SplaTAM[16]	35.5	36.1	149.7	91.2	12.5	19.0	57.4
GS-SLAM[17]	37.5	26.8	46.8	50.4	31.9	4.8	33.1
DROID-VO[19]	5.4	4.6	21.4	46.0	8.9	5.9	15.4
DG-SLAM(Ours)	**3.7**	**4.1**	**4.5**	**6.9**	10.0	**3.5**	**5.5**

Table 3: **Camera tracking results on several dynamic scene sequences in the *BONN* dataset.** "$*$" denotes the version reproduced by NICE-SLAM. "X" denotes the tracking failures. The metric is ATE and the unit is [cm].

4.1 Evaluation of generating motion mask

We evaluated our method on the `balloon` and `move_no_box2` sequences of the *BONN* dataset to show the qualitative results of the generated motion mask. In these two sequences, in addition to the typical motion of pedestrians, there are movements of atypical objects accompanying the human, such as balloons and boxes. It might be missed if we rely solely on the semantic prior. As shown in Fig. 2, our generated methods notably improve the precision of motion mask segmentation, effectively reducing the inconsistencies along edge regions and detecting the true dynamic objects.

4.2 Evaluation of mapping performance

To more effectively showcase the performance of our system in dynamic environments, we evaluate the reconstruction results from both qualitative and quantitative perspectives. Given dynamic scene datasets seldom offer static GT mesh or point cloud, we utilize the *BONN* dataset for our quantitative analytical experiments. We compare our DG-SLAM method against current state-of-the-art neural-based SLAM methods, all of which are available as open-source projects.

As shown in Tab. 1, our method significantly surpasses contemporary approaches in terms of accuracy, completion, and completion ratio metrics, achieving state-of-the-art performance. Meanwhile, the reconstructed static map can be rendered with high fidelity, as shown in Fig. 3. This also indirectly demonstrates that our methods can generate a more accurate static map compared with other mainstream SLAM systems.

4.3 Evaluation of tracking performance

To comprehensively assess the tracking performance of our DG-SLAM approach, we conduct comparative analyses within highly dynamic, slightly dynamic, and static environments. The comparison methods encompass classical SLAM methods such as ORB-SLAM3 [3], ReFusion [5],MID-fusion [43], and EM-fusion [44], and widely recognized NeRF-based SLAM systems such as NICE-SLAM [7], iMap [6], Vox-Fusion [10], ESLAM [9], Co-SLAM [8]. We also incorporate a comparison with the newly proposed dynamic neural RGB-D SLAM system, Rodyn-SLAM [33]. Furthermore, our evaluation extends to the latest advancements in 3D Gaussian-based SLAM, including SplaTAM [16] and GS-SLAM [17].

Dynamic scenes. As shown in Tab. 2, we report the results on three highly dynamic sequences, two slightly dynamic sequences, and one static sequence from the TUM RGB-D dataset. Our system exhibits exceptional tracking performance, attributed to the implementation of the map point deleting strategy and the powerful coarse-to-fine camera tracking algorithm. Furthermore, the tracking capabilities of our system have also been rigorously evaluated on the intricate and demanding BONN RGB-D dataset, with outcomes presented in Tab. 3. In dynamic scenarios characterized by heightened complexity and challenge, our method has consistently demonstrated superior performance, underscoring its effectiveness and reliability in real-world navigation applications. Meanwhile, we also showcased the number of iterations of the tracking and mapping process, taking TUM as an example, as shown in Tab. 5. Compared to other methods, our method achieved superior results while maintaining competitive iterations and efficiency.

Static scenes. To better illustrate the robustness of our system, we also evaluate our methods with existing SLAM on common real-world static sequences from ScanNet [39]. As shown in Tab. 4, our DG-SLAM still achieves competitive performance within static scenes with fewer tracking and mapping iterations, despite our method being designed for dynamic scenes. Notably, the motion mask will become irrelevant in static scenes. Thus, it can sufficiently demonstrate the effectiveness of our proposed hybrid camera tracking strategy and adaptive Gaussian point management strategy.

Method	00	59	106	169	181	207	Avg.
NICE-SLAM [7]	12.0	14.0	7.9	10.9	13.4	6.2	10.7
Co-SLAM [8]	**7.1**	11.1	9.4	**5.9**	11.8	**7.1**	8.8
Point-SLAM [31]	10.2	7.8	8.7	22.2	14.8	9.5	12.2
SplaTAM [16]	12.8	10.1	17.7	12.1	11.1	7.5	11.9
GS-SLAM [17]	13.6	**7.6**	8.1	13.7	34.6	12.5	15.1
DG-SLAM(Ours)	7.9	11.5	**8.0**	8.3	**7.3**	8.2	**8.6**

Table 4: **Camera tracking results on *ScanNet*.** The metric is ATE and the unit is [cm].

Iterations	tracking	mapping
NICE-SLAM [7]	200	60
Co-SLAM [8]	**20**	**30**
ESLAM [9]	200	60
Point-SLAM [31]	200	150
SplaTAM [16]	200	**30**
GS-SLAM [17]	100	81
DG-SLAM(Ours)	**20**	40

Table 5: **Running iterations on *TUM*.**

4.4 Ablation study

To evaluate the efficacy of the proposed algorithm in our system, we conducted ablation studies across seven representative sequences from the *BONN* dataset. Given that the semantic priors within the *TUM* dataset have covered major motion categories (humans) while motion categories in *BONN* contained undefined dynamic objects such as balloons and boxes, thus we opted to perform ablation studies on the *BONN* dataset. We calculated the average ATE and STD metrics to illustrate the impact of various components on the overall system performance. As the results presented in Tab. 6, the findings affirm the effectiveness of all proposed methods in improving camera tracking capabilities. Specifically, the strategies for adding and pruning points are equally crucial, significantly impacting the quality of the Gaussian map reconstruction and, in turn, affecting the tracking performance. The depth warp operation effectively removes floaters from multiple viewpoints, thereby noticeably

https://doi.org/10.52202/079017-1633

enhancing the quality of the rendered images. One of the main contributions comes from the hybrid camera tracking strategy, which indirectly underscores the importance of eliminating inconsistencies between pose estimation and map reconstruction.

4.5 Time consumption analysis

As shown in Tab. 7, we report time consumption (per frame) of the tracking and mapping without computing semantic segmentation. These results were achieved through an identical experimental setup that involved conducting 20 iterations for tracking and 40 iterations for mapping, all processed on an RTX 3090Ti GPU. Benefiting from the rapid execution speed of Droid-VO and fast rendering of 3D Gaussian Splatting, our method exhibits a leading edge in terms of time consumption during the tracking process. Although our approach does not match the mapping speed of Co-SLAM, it achieves high-quality mapping and high-fidelity rendering with a competitive mapping running time. We also evaluate the inference time of our used semantic segmentation network, which required 163ms for every frame. It should be noted that our approach does not focus on the specific semantic segmentation network used, but rather on the fusion method itself.

	ATE[cm]↓	STD[cm]↓
w/o Add	6.63	3.08
w/o Prune	6.89	3.16
w/o Dep. Warp	6.40	3.22
w/o Seg	15.27	7.47
w/o Fine Tracking	7.36	3.62
DG-SLAM(Ours)	**5.51**	**2.79**

Table 6: **Ablation study on *BONN* dataset.**

	Tracking [ms]↓	Mapping [ms]↓	Avg. Running [ms] ↓
NICE-SLAM [7]	3186.2	1705.1	4892.3
ESLAM [9]	2045.9	1641.4	3688.5
Point-SLAM [31]	2279.5	1544.4	3823.9
Co-SLAM [8]	101.4	**140.1**	**241.6**
DG-SLAM(Ours)	**89.2**	549.3	645.9

Table 7: **Run-time comparison on *TUM* f3/w_s.**

5 Conclusion

In this paper, we have presented DG-SLAM, a robust dynamic Gaussian splatting SLAM with hybrid pose optimization under dynamic environments. Via motion mask filter strategy and coarse-to-fine camera tracking algorithm, our system significantly advances the accuracy and robustness of pose estimation within dynamic scenes. The proposed adaptive 3D Gaussians adding and pruning strategy effectively improves the quality of reconstructed maps and rendering images. We demonstrate its effectiveness in achieving state-of-the-art results in camera pose estimation, scene reconstruction, and novel-view synthesis in dynamic environments. While the tracking and reconstruction of large-scale scenes is currently the biggest limitation of our system, we believe it will be addressed by a more flexible loop-closure optimization algorithm in future work. Moreover, the accuracy of pose estimation of our system is still influenced by the segmentation precision of the semantic prior. Therefore, efficiently perceiving moving objects within the dynamic scenes remains an unresolved issue that warrants further exploration.

Acknowledgement

This work is supported by National Key R&D Program of China (No.2019YFA0709502, No.2018YFC1312904), National Natural Science Foundation of China (Grant No.62106050 and 62376060), Natural Science Foundation of Shanghai (Grant No.22ZR1407500), ZJ Lab, Shanghai Center for Brain Science and Brain-Inspired Technology, and NIO University Programme (NIO UP).

References

[1] John J Leonard and Hugh F Durrant-Whyte. Simultaneous map building and localization for an autonomous mobile robot. In *IROS*, 1991. 1

[2] Raul Mur-Artal and Juan D Tardós. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE TRO*, 2017. 1, 2

[3] Carlos Campos, Richard Elvira, Juan J Gómez Rodríguez, José MM Montiel, and Juan D Tardós. Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam. *IEEE TRO*, 2021. 1, 2, 8, 9

[4] Haochen Jiang, Rui Qian, Liang Du, Jian Pu, and Jianfeng Feng. Ul-slam: A universal monocular line-based slam via unifying structural and non-structural constraints. *IEEE TASE*, 2024. 1

[5] Emanuele Palazzolo, Jens Behley, Philipp Lottes, Philippe Giguere, and Cyrill Stachniss. Refusion: 3d reconstruction in dynamic environments for rgb-d cameras exploiting residuals. In *IROS*, 2019. 1, 2, 4, 6, 8, 9

[6] Edgar Sucar, Shikun Liu, Joseph Ortiz, and Andrew Davison. iMAP: Implicit mapping and positioning in real-time. In *ICCV*, 2021. 1, 3, 8, 9

[7] Zihan Zhu, Songyou Peng, Viktor Larsson, Weiwei Xu, Hujun Bao, Zhaopeng Cui, Martin R. Oswald, and Marc Pollefeys. Nice-slam: Neural implicit scalable encoding for slam. In *CVPR*, 2022. 1, 3, 7, 8, 9, 10

[8] Hengyi Wang, Jingwen Wang, and Lourdes Agapito. Co-slam: Joint coordinate and sparse parametric encodings for neural real-time slam. In *CVPR*, 2023. 1, 3, 7, 8, 9, 10

[9] Mohammad Mahdi Johari, Camilla Carta, and François Fleuret. Eslam: Efficient dense slam system based on hybrid representation of signed distance fields. In *CVPR*, 2023. 1, 3, 7, 8, 9, 10

[10] Xingrui Yang, Hai Li, Hongjia Zhai, Yuhang Ming, Yuqian Liu, and Guofeng Zhang. Vox-fusion: Dense tracking and mapping with voxel-based neural implicit representation. In *ISMAR*, 2022. 1, 3, 8, 9

[11] Richard A Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J Davison, Pushmeet Kohi, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *ISMAR*, 2011. 1

[12] Thomas Schöps, Torsten Sattler, and Marc Pollefeys. Surfelmeshing: Online surfel-based mesh reconstruction. *IEEE TPAMI*, 2019. 1

[13] Thomas Schops, Torsten Sattler, and Marc Pollefeys. Bad slam: Bundle adjusted direct rgb-d slam. In *CVPR*, 2019. 1

[14] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Commun Acm*, 2021. 1

[15] Chi Yan, Delin Qu, Dong Wang, Dan Xu, Zhigang Wang, Bin Zhao, and Xuelong Li. Gs-slam: Dense visual slam with 3d gaussian splatting. In *CVPR*, 2024. 2, 3

[16] Nikhil Keetha, Jay Karhade, Krishna Murthy Jatavallabhula, Gengshan Yang, Sebastian Scherer, Deva Ramanan, and Jonathon Luiten. Splatam: Splat, track & map 3d gaussians for dense rgb-d slam. In *CVPR*, 2024. 2, 3, 7, 8, 9

[17] Hidenobu Matsuki, Riku Murai, Paul H. J. Kelly, and Andrew J. Davison. Gaussian Splatting SLAM. In *CVPR*, 2024. 2, 3, 7, 8, 9

[18] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM TOG*, 2023. 2, 3, 4

[19] Zachary Teed and Jia Deng. DROID-SLAM: Deep Visual SLAM for Monocular, Stereo, and RGB-D Cameras. In *NeurIPS*, 2021. 2, 5, 7, 8

[20] Chao Yu, Zuxin Liu, Xin-Jun Liu, Fugui Xie, Yi Yang, Qi Wei, and Qiao Fei. Ds-slam: A semantic visual slam towards dynamic environments. In *IROS*, 2018. 2

[21] Berta Bescos, José M Fácil, Javier Civera, and José Neira. Dynaslam: Tracking, mapping, and inpainting in dynamic scenes. *IEEE RAL*, 2018. 2

 https://doi.org/10.52202/079017-1633

[22] Linhui Xiao, Jinge Wang, Xiaosong Qiu, Zheng Rong, and Xudong Zou. Dynamic-slam: Semantic monocular visual localization and mapping based on deep learning in dynamic environment. *IEEE RAS*, 2019. 2

[23] Berta Bescos, Carlos Campos, Juan D Tardós, and José Neira. Dynaslam ii: Tightly-coupled multi-object tracking and slam. *IEEE RAL*, 2021. 2

[24] Jun Zhang, Mina Henein, Robert Mahony, and Viorela Ila. Vdo-slam: a visual dynamic object-aware slam system. *arXiv preprint*, 2020. 2

[25] João Carlos Virgolino Soares, Marcelo Gattass, and Marco Antonio Meggiolaro. Crowd-slam: visual slam towards crowded environments using object detection. *Journal of Intelligent & Robotic Systems*, 2021. 2

[26] Mingrui Li, Jiaming He, Guangan Jiang, and Hongyu Wang. Ddn-slam: Real-time dense dynamic neural implicit slam with joint semantic encoding. *arXiv preprin*, 2024. 2

[27] Shihao Shen, Yilin Cai, Wenshan Wang, and Sebastian Scherer. Dytanvo: Joint refinement of visual odometry and motion segmentation in dynamic environments. In *ICRA*, 2023. 2

[28] Weicai Ye, Xingyuan Yu, Xinyue Lan, Yuhang Ming, Jinyu Li, Hujun Bao, Zhaopeng Cui, and Guofeng Zhang. Deflowslam: Self-supervised scene motion decomposition for dynamic dense slam. *arXiv preprint*, 2022. 2

[29] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Commun ACM*, 2021. 2

[30] Jiahui Huang, Shi-Sheng Huang, Haoxuan Song, and Shi-Min Hu. Di-fusion: Online implicit 3d reconstruction with deep priors. In *CVPR*, 2021. 3

[31] Erik Sandström, Yue Li, Luc Van Gool, and Martin R. Oswald. Point-slam: Dense neural point cloud-based slam. In *ICCV*, 2023. 3, 6, 9, 10

[32] Qiangeng Xu, Zexiang Xu, Julien Philip, Sai Bi, Zhixin Shu, Kalyan Sunkavalli, and Ulrich Neumann. Point-nerf: Point-based neural radiance fields. In *CVPR*, 2022. 3

[33] Haochen Jiang, Yueming Xu, Kejie Li, Jianfeng Feng, and Li Zhang. Rodyn-slam: Robust dynamic dense rgb-d slam with neural radiance fields. *RAL*, 2024. 3, 8, 9

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

[35] Tianchen Deng, Yaohui Chen, Leyan Zhang, Jianfei Yang, Shenghai Yuan, Danwei Wang, and Weidong Chen. Compact 3d gaussian splatting for dense visual slam. *arXiv preprint*, 2024. 3

[36] Georg Klein and David Murray. Parallel tracking and mapping for small ar workspaces. In *ISMAR*, 2007. 3

[37] Richard A. Newcombe, Steven J. Lovegrove, and Andrew J. Davison. Dtam: Dense tracking and mapping in real-time. In *ICCV*, 2011. 3

[38] Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. A benchmark for the evaluation of rgb-d slam systems. In *IROS*, 2012. 6, 7

[39] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *CVPR*, 2017. 6, 9

[40] Berthold KP Horn. Closed-form solution of absolute orientation using unit quaternions. *Josa a*, 1987. 7

[41] Jitesh Jain, Jiachen Li, MangTik Chiu, Ali Hassani, Nikita Orlov, and Humphrey Shi. OneFormer: One Transformer to Rule Universal Image Segmentation. In *CVPR*, 2023. 7

[42] Martin Rünz and Lourdes Agapito. Co-fusion: Real-time segmentation, tracking and fusion of multiple objects. In *ICRA*, 2017. 8

[43] Binbin Xu, Wenbin Li, Dimos Tzoumanikas, Michael Bloesch, Andrew Davison, and Stefan Leutenegger. Mid-fusion: Octree-based object-level multi-instance dynamic slam. In *ICRA*, 2019. 8, 9

[44] Michael Strecke and Jorg Stuckler. Em-fusion: Dynamic object-level slam with probabilistic data association. In *ICCV*, 2019. 8, 9

 https://doi.org/10.52202/079017-1633

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: We have clearly stated the contribution and scope of the paper in the abstract.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: We have discussed the limitations of the work in the conclusion part.

   Guidelines:

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

3. **Theory Assumptions and Proofs**

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

   Answer: [Yes]

Justification: We have provided the theoretical assumption and proof of this paper in the method part.

Guidelines:

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

4. **Experimental Result Reproducibility**

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

Answer: [Yes]

Justification: We have disclosed all the information needed to reproduce the main experimental results of the paper in the experiment part.

Guidelines:

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

5. **Open access to data and code**

https://doi.org/10.52202/079017-1633

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

Answer: [No]

Justification: We have not released the code of this paper in the submission.

Guidelines:

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

6. **Experimental Setting/Details**

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

Answer: [Yes]

Justification: We have specified all the training and test details of this paper in the experiment part.

Guidelines:

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

7. **Experiment Statistical Significance**

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

Answer: [Yes]

Justification: We have reported appropriate information about the statistical significance of the experiments in the experiment part.

Guidelines:

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

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

8. **Experiments Compute Resources**

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

   Answer: [Yes]

   Justification: We have provided sufficient information on the computer resources of this method in the experiment part.

   Guidelines:

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

9. **Code Of Ethics**

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

   Answer: [Yes]

   Justification: We have conducted research under the NeurIPS Code Of Ethics.

   Guidelines:

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

10. **Broader Impacts**

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

    Answer: [Yes]

    Justification: We have discussed the societal impacts of this paper in the introduction and method parts.

    Guidelines:

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

https://doi.org/10.52202/079017-1633

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

11. **Safeguards**

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

    Answer: [NA]

    Justification: This paper has not released data or models that have a high risk for misuse.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: All the assets used in the paper are properly licensed. Our paper cites the original paper that produced the code package or dataset.

    Guidelines:

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

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

13. **New Assets**

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

    Answer: [NA]

    Justification: Our paper does not release new assets, including data and models. We also did not publish or submit the code.

    Guidelines:

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

14. **Crowdsourcing and Research with Human Subjects**

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

    Answer: [NA]

    Justification: The research topic of this paper is robotic SLAM, not involving crowdsourcing nor research with human subjects.

    Guidelines:

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

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

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

    Answer: [NA]

    Justification: The research topic of this paper is robotic SLAM, not involving crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

https://doi.org/10.52202/079017-1633

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.