Uncertainty-Driven 3D Gaussian Splatting for Robust Real-Time RGB-D SLAM

Anonymous Authors

***Abstract*—3D Gaussian Splatting has been a promising module for autonomous robots to perform localization while reconstructing a photorealistic environment. However, its robustness in real-world deployments is limited. The lack of effective noise mitigation and efficient loop closure leads to the accumulation of errors from motion blur and sensor noise. This paper presents μSLAM, a robust, real-time RGB-D Simultaneous Localization and Mapping (SLAM) system designed to address these limitations through a synergistic, uncertainty-driven framework. The system is built upon a hybrid map that integrates a sparse feature-based backbone with an uncertainty-aware Gaussian field. Local noise is mitigated through an information-theoretic strategy for keyframe selection, coupled with a confidence-modulated adaptive optimizer for map refinement. Global consistency is ensured by a novel coarse-to-fine loop closure process. This process leverages the embedded sparse map for robust initial alignment and the dense field's rendering capabilities for high-precision refinement. Extensive experiments on public benchmarks and real-world robotic sequences demonstrate that μSLAM achieves state-of-the-art tracking accuracy and reconstruction quality. The system exhibits superior robustness to noise and drift. Crucially, it is the only evaluated system to deliver this high level of performance while consistently operating in real-time (over 30 frames per second), making it a comprehensive solution for practical robotic applications.**

***Note to Practitioners*—This paper is motivated by the dual demand in indoor service robotics and AR/VR industries for creating high-fidelity, photorealistic 3D maps while maintaining precise, globally consistent self-localization in real-time. A major practical bottleneck is the reliance on professional stabilization equipment; using low-cost cameras on mobile robots or handheld devices often results in trajectory drift and map corruption caused by rapid motion blur and camera shake. This work addresses these issues by introducing μSLAM, a system that ensures robust pose estimation and flexible data collection in noisy environments. Our solution improves system performance in three key aspects: 1) Efficiency: By quantifying map "confidence," the system intelligently selects only the most informative frames for processing; 2) Quality: It employs targeted optimization strategies to re-construct only the regions that are not fully modeled; 3) Reliability: It integrates a sparse geometric backbone with loop closure mechanisms to guarantee accurate, drift-free trajectory estimation for both autonomous robots and human operators, even during aggressive motion. The primary contribution is a quantified approach to handling 3DGS noise, achieving an optimal balance between real-time performance and robustness. A current limitation is the hardware requirement (high-end GPUs like RTX 3090). Future work will explore better theoretical confidence algorithms to enable deployment on resource-constrained platforms and extend the system to Active SLAM. Potential applications include providing reliable pose and confidence data for robotic grasping and navigation, as well as constructing realistic scenes for training Vision-Language-Action (VLA) models.**

***Index Terms*—Simultaneous localization and mapping, 3d gaussian splatting, uncertainty modeling, service robots, visual-based navigation.**

I. Introduction

Simultaneous Localization and Mapping (SLAM) is a fundamental capability for autonomous robots, enabling them to estimate their pose while concurrently building a map of an unknown environment. The ability to reconstruct high-fidelity, photorealistic environments is crucial for advanced robotic tasks such as navigation [1]–[3], manipulation [4], and human-robot interaction [5], [6]. With recent advancements in GPU computing, neural rendering techniques like 3D Gaussian Splatting (3DGS) [7] have emerged as a compelling foundation for SLAM systems, offering a unique combination of high rendering quality, flexible scalability, and real-time rendering speed that is highly suitable for robotic applications.

However, the integration of 3DGS into robust, real-world SLAM systems faces significant challenges that limit its practical deployment. Existing 3DGS-SLAM methods are hindered by two primary, interconnected problems: inefficient noise mitigation and a lack of robust global consistency mechanisms. *Firstly, regarding local noise mitigation,* most approaches [8]–[10] adopt the vanilla 3DGS optimization, which treats all observational data uniformly. This makes them highly susceptible to sensor noise and motion blur, which are unavoidable in real-world robotics. Corrupted data leads to incorrect gradients, which not only degrades map quality but also introduces short-term tracking drift. Furthermore, naive keyframe selection strategies, often based on co-visibility, lead to the processing of redundant frames, wasting computational resources and exacerbating the impact of noisy observations over time.

Secondly, concerning global consistency, the frame-to-model tracking inherent to these systems inevitably leads to long-term drift. While some methods incorporate loop closure, their solutions are often suboptimal for real-time robotics. They typically rely on computationally expensive submap management and registration schemes [11], or they offload the task to external, uninterpretable large vision models [12], introducing significant overhead and creating a dependency on black-box systems.

In this paper, we present μSLAM, a lightweight, multi-stage system whose name encapsulates its core principles: it is a **M**ulti-stage, **U**ncertainty-driven framework (hence MU, or μ), that uniquely models **micro**-scopic, per-primitive uncertainty. To mitigate local noise, we introduce the *Uncertainty-Aware Gaussian Field (UGF)*, a novel map representation that explicitly models the confidence of each Gaussian primitive, as shown in Fig. 1. This confidence is then leveraged throughout our framework: an information-theoretic keyframe selection strategy, based on Shannon mutual information, ensures that only the most informative new views are processed,

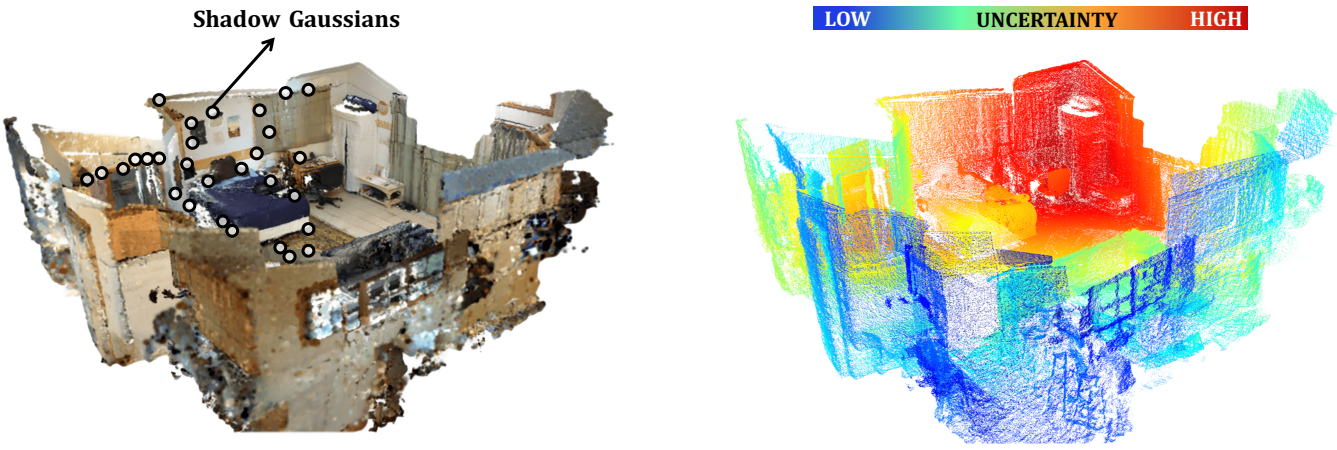

Figure 1. **The synergy of our hybrid, uncertainty-aware map.** This figure illustrates our core map representation, which combines a photorealistic 3DGS field (left) with a per-primitive confidence visualization (right). The embedded **shadow Gaussians** (white circles) provide a globally consistent geometric backbone. The **confidence map** (blue: low, red: high) explicitly models the system's belief about the reconstruction's stability. This synergistic approach, combining geometric features with an uncertainty-driven dense field, is the key to our system's superior balance of robustness and real-time performance.

while a confidence-modulated adaptive optimizer, grounded in stochastic optimization theory, intelligently allocates computational resources to the most uncertain regions of the map. To ensure global consistency, we propose a synergistic *coarse-to-fine loop closure* process. This is enabled by embedding a sparse feature map as "*shadow Gaussians*" within the UGF, as shown in Fig. 1, providing a robust geometric backbone for coarse loop detection and initial pose estimation. This is followed by a high-precision pose refinement that leverages the novel view synthesis capabilities of the dense 3D Gaussian Splatting field.

The main contributions of this paper are:

- **A novel map representation**, the Uncertainty-Aware Gaussian Field (UGF), that extends 3D Gaussian Splatting with an explicit, per-primitive confidence property derived from optimization dynamics.
- **An uncertainty-aware framework** for local noise mitigation, comprising an information-theoretic keyframe selection strategy and a confidence-modulated adaptive optimizer to ensure both efficiency and robustness.
- **A synergistic coarse-to-fine loop closure mechanism** built upon a unified sparse-dense map, which achieves robust and high-precision global relocalization without reliance on external models or heavy submap management.

In summary, we make three key claims, which are validated by our extensive experimental evaluation:

1) **Superior Tracking Accuracy:** μSLAM achieves state-of-the-art tracking accuracy, outperforming existing rendering-based SLAM approaches.
2) **High-Fidelity Reconstruction:** Our approach delivers high-fidelity reconstruction quality that is on par with, or superior to, competing methods, particularly in challenging real-world scenes.
3) **Robustness with Real-Time Efficiency:** μSLAM operates at a consistent real-time frame rate (>30 FPS) and successfully completes challenging sequences where other high-fidelity systems fail, demonstrating a superior balance between speed, robustness, and quality.

The source code will be made publicly available upon the publication of this article.

II. Related Work

Our work proposes a synergistic framework that combines the strengths of sparse-feature SLAM and map-centric RGB-D SLAM to achieve robust global consistency and mitigate noise accumulation. Accordingly, this section first reviews the state-of-the-art within each of these two primary categories and then discusses the emerging class of hybrid methods that seek to integrate them.

A. Sparse-Feature SLAM

This subsection reviews sparse-feature SLAM, focusing on its strategies for mitigating noise and achieving global consistency, before analyzing the inherent limitations of these approaches.

Sparse-feature SLAM excels at achieving accurate pose tracking and globally consistent map estimation. Its advantages stem primarily from two key properties. First, by representing the environment with a sparse set of handcrafted or learned features, systems like ORB-SLAM3 [13] enable efficient and accurate pose estimation through robust multi-stage data association.

Second, and critically for global consistency, the sparse feature map is inherently suited for a highly efficient loop closure process. This process typically involves descriptor-based place recognition (e.g., DBoW2 [14]) to identify previously visited locations, followed by geometric verification and global Bundle Adjustment (BA) [15]. The map's inherent sparsity is crucial, as it allows the BA to rapidly propagate corrections and mitigate cumulative drift across the entire trajectory.

However, the primary limitation of sparse-feature SLAM is that its map representation, while efficient for localization,

lacks the density and photometric detail required for high-level spatial intelligence tasks such as photorealistic rendering, dense navigation, and rich scene understanding.

B. Map-Centric 3DGS-based SLAM

In contrast to sparse-feature SLAM, which prioritizes localization, map-centric systems aim to build dense 3D representations suitable for higher-level tasks. The evolution of these representations has moved from traditional explicit structures like Truncated Signed Distance Functions (TSDFs) [16]–[21], Octomaps [22], [23], or surfels [24]–[26], to modern neural implicit representations. The advent of Neural Radiance Fields (NeRFs) introduced a paradigm shift toward high-fidelity modeling. While subsequent work improved scalability with composite representations such as hierarchical voxel grids [27], octrees [28], spatial hashing [29], tri-plane grids [30], [31], or unordered points [32]–[34], the high computational cost of volumetric rendering remained a significant bottleneck for real-time SLAM.

The recent introduction of 3DGS [7] has revolutionized the field. By using an explicit, optimizable primitive-based representation, 3DGS achieves rendering quality that is on par with, or even superior to, NeRF, while being orders of magnitude faster to train and render. This unique combination of speed and fidelity makes it a compelling choice for the core map representation in modern SLAM systems. Consequently, a new class of "*coupled*" 3DGS-SLAM systems [35]–[40] has emerged, which use the 3DGS field as the sole map representation for both tracking and mapping via render-based optimization.

Despite their high fidelity, these coupled systems face two fundamental challenges rooted in their direct reliance on dense photometric data. First, their *optimization efficiency* is substantially lower than that of sparse SLAM, exacerbated by a lack of tailored keyframe selection criteria and targeted optimization strategies. Second, they are highly susceptible to *local error accumulation*, as they treat all observations uniformly, allowing sensor noise and motion blur to degrade both the map and the pose estimates. The lack of a principled way to quantify and mitigate this uncertainty—a concept well-established in classical robotics—is a primary weakness of current approaches.

The challenge of managing uncertainty is a cornerstone of classical probabilistic robotics [41], where methods such as Kalman filters and particle filters primarily model the uncertainty of the robot's pose within a given map. However, as map representations have evolved from sparse landmarks to millions of dense primitives, the uncertainty inherent in the *map itself* has become a dominant factor affecting overall system robustness. These two forms of uncertainty—pose and map—are deeply intertwined: a map with high, unmodeled uncertainty leads to ambiguous localization, while inaccurate pose estimates corrupt the map, further increasing its uncertainty. Our work addresses this gap by extending the principles of probabilistic state estimation from the robot's trajectory to the individual components of the dense map. By explicitly modeling a per-primitive confidence, we create a map that is not only a geometric and photometric representation but also an "uncertainty-aware" participant in the state estimation process, a crucial step for achieving robust performance in real-world conditions. The subsequent section will review hybrid methods that attempt to address global error accumulation.

C. Hybrid 3DGS-SLAM and Global Consistency

To address the limitations of coupled systems, a promising direction is the development of *hybrid* (or "*decoupled*") SLAM frameworks that combine a sparse-feature SLAM front-end with a dense 3DGS back-end. This section reviews such approaches and analyzes the specific challenges of integrating robust loop closure to achieve global consistency.

The core idea of a hybrid framework is to use a sparse SLAM front-end to provide high-quality initial poses for the 3DGS back-end's optimization. While this improves short-term tracking robustness, it does not inherently solve long-term drift. True global consistency requires a dedicated loop closure mechanism, but integrating one into a hybrid 3DGS system presents two fundamental difficulties [42]–[44]. First, *achieving precise relative pose estimation* between a current frame and a distant loop candidate is challenging. Second, after a loop is closed, *efficiently propagating these corrections* to the dense 3DGS global map without distortion is non-trivial.

Prior works have attempted to address these integration challenges with distinct strategies. LoopSplat [45] tackles the problem by tracking keyframes against local submaps. However, this approach introduces significant computational overhead from submap management and merging, and it struggles to close large loops that lack good initial pose estimates. Conversely, 2DGS-SLAM [12] bypasses the intrinsic relocalization problem by offloading both loop detection and pose estimation to an external large visual model (MAST3R [46]). While this avoids submap overhead, it fails to exploit the 3DGS map's own localization capabilities, incurs additional costs, and makes the system's robustness entirely dependent on a black-box model.

The limitations of current hybrid approaches highlight a clear gap in the literature: the need for a deeply integrated, synergistic framework that can achieve robust and efficient end-to-end loop correction without relying on heavyweight submaps or external models. Our work is designed to fill this gap by proposing a unified sparse-dense map representation that enables such a pipeline.

III. Method

This section details the architecture and core components of our proposed system, μSLAM. We begin with a high-level overview of the system's structure and unified map representation, followed by detailed explanations of each key module.

A. System Overview

The architecture of our proposed system, μSLAM, is depicted in Fig. 2. It is a multi-stage, uncertainty-driven framework designed to achieve robust, real-time, and globally

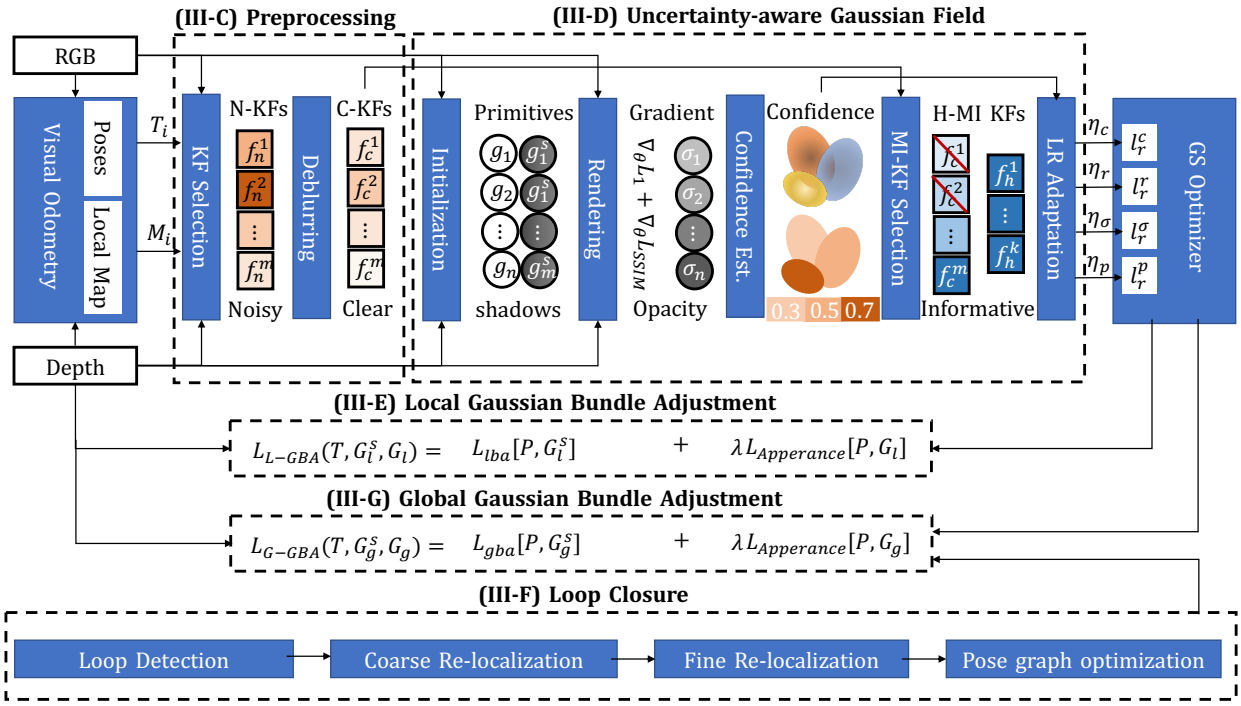

Figure 2. **The complete architecture of our μSLAM system.** Our system is a multi-stage, uncertainty-driven pipeline that synergistically integrates three core stages: (1) a pre-processing front-end to safeguard data integrity, (2) an uncertainty-aware back-end that builds and refines the core map, and (3) integrated noise mitigation and loop closure mechanisms to ensure local and global consistency.

consistent 3DGS-based SLAM. The core of our approach is a *synergistic integration* of a sparse geometric backbone with a dense, uncertainty-aware photometric map. To achieve this, we introduce a unified map representation that embeds a sparse feature point cloud from a visual odometry front-end directly into our map as a set of special primitives termed *shadow Gaussians*. These primitives are rendered invisible to the photometric loss but retain their 3D positions for geometric tasks.

More specifically, our unified map representation M is the union of the set of uncertainty-aware Gaussians, $\mathcal{G}_U$, and the set of shadow Gaussians, $\mathcal{G}_S$:

$$\begin{aligned} M &= \mathcal{G}_U \cup \mathcal{G}_S, \\ \mathcal{G}_U &= \{g_i^U(\boldsymbol{\theta}_i^U, C_i)\}_{i=1}^{N_U}, \\ \mathcal{G}_S &= \{g_j^S(\boldsymbol{\mu}_j)\}_{j=1}^{N_S}, \end{aligned} \tag{1}$$

where N_U and N_S denote the total number of uncertainty-aware and shadow primitives, respectively. Each uncertainty-aware Gaussian g_i^U is composed of its learnable parameters and its confidence property. The vector $\boldsymbol{\theta}_i^U = \{\mathbf{c}_i, r_i, \alpha_i\}$ contains the learnable parameters for an isotropic Gaussian (color, radius, opacity), inspired by [9]. The item $C_i \in [0, 1]$ denotes the *confidence property*, which is central to our framework and is used to guide keyframe selection and modulate learning rates.

The set $\mathcal{G}_S$ contains the shadow Gaussians, where each g_j^S is primarily defined by its 3D position $\boldsymbol{\mu}_j$, with its other parameters (opacity α_s, radius r_s) fixed at near-zero values. This position $\boldsymbol{\mu}_j$ is directly inherited from a corresponding feature point in the sparse map. These primitives serve as the geometric anchors for robust tracking and global optimization.

Our system pipeline, guided by this unified map, consists of four main modules that operate sequentially and in parallel:

1) **Pre-processing** (Sec. III-B): Safeguards data integrity by filtering incoming frames both geometrically and photometrically.
2) **Uncertainty-Aware Optimization** (Sec. III-C): Forms the core of our local noise mitigation by quantifying per-primitive confidence, selecting keyframes based on mutual information, and dynamically adapting learning rates.
3) **Tracking and Local BA** (Sec. III-D): Performs robust per-frame tracking and refines the local map via our novel, confidence-aware Local Gaussian Bundle Adjustment (LGBA).
4) **Loop Closure and Global Consistency** (Sec. III-E): Eliminates long-term drift through a coarse-to-fine loop correction and a two-stage global optimization.

The subsequent sections will provide a comprehensive explanation of each component.

B. Pre-processing

The robustness of any SLAM system is fundamentally dependent on the quality of its input data. This is particularly true for 3DGS-based SLAM, which employs dense photometric supervision for both pose and map optimization, making it highly sensitive to sensor observations. Raw sensor streams from robotic platforms, often corrupted by motion blur and noise, can introduce significant uncertainty that degrades

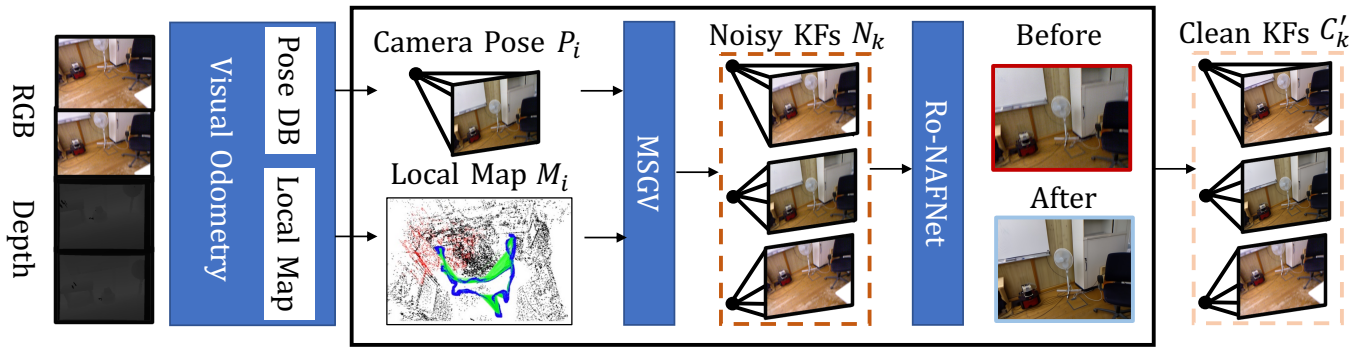

Figure 3. **Overview of our two-stage pre-processing pipeline.** The raw RGB-D stream is first processed by a **Geometric Stability Filter** (MSGV) to select a sparse set of geometrically stable, yet potentially noisy, keyframes ($\mathbf{N}_k$). These are then enhanced by our robotics-optimized NAFNet (**Ro-NAFNet**) to produce a final, clean set of keyframes ($\mathbf{C}_k$) for the SLAM back-end.

the convergence and accuracy of the entire state estimation process. The primary goal of our pre-processing module is therefore to safeguard the integrity of this supervisory data stream through a principled, two-stage filtering pipeline.

As illustrated in Fig. 3, the pipeline begins by mitigating *geometric instability*. This first stage is handled by our *Geometric Stability Filter*, labeled MSGV in the figure. To ensure that only informative and robustly trackable frames are processed, we apply a rigorous geometric stability filter. A newly acquired frame F_k with pose $\mathbf{T}_k \in SE(3)$ is designated a stable candidate keyframe ($\mathbf{N}_k$) only if it satisfies three criteria: a sufficient translational baseline, robust tracking continuity, and the provision of novel visual information. We formalize these conditions with respect to the last selected reference frame, F_{ref}. Let $\mathbf{T}_{k,ref} = \mathbf{T}_k\mathbf{T}_{ref}^{-1}$ be the relative transformation, and N_k be the number of features successfully tracked from F_{ref}. The conditions for candidacy are:

$$\|\text{trans}(\mathbf{T}_{k,ref})\| > \tau_{\text{trans}}, \tag{2}$$

$$N_k > \tau_{\text{track}}, \tag{3}$$

$$\frac{|\mathcal{M}_k \cap \mathcal{M}_{ref}|}{|\mathcal{M}_k|} < \tau_{\text{overlap}}, \tag{4}$$

where $\text{trans}(\cdot)$ extracts the translation vector, and $\mathcal{M}_k$ and $\mathcal{M}_{ref}$ are the sets of map points observed by their respective frames. The hyperparameters τ_{trans}, τ_{track}, and τ_{overlap} represent the thresholds for minimum translation, minimum tracked features, and maximum feature overlap, respectively. Collectively, they ensure sufficient parallax for triangulation, reliable frame-to-frame motion estimation, and data non-redundancy. Critically, this filter acts as a gatekeeper, passing only a sparse fraction of geometrically valuable frames (typically $< 15\%$ of the input stream) to the subsequent enhancement stage.

The second stage tackles *photometric degradation*. Each geometrically stable candidate's color image, $\mathbf{C}_k$, is enhanced by our robotics-optimized NAFNet (*Ro-NAFNet*), a neural network Φ based on [47]. This network, pre-trained on robotic vision datasets, operates on the noisy and potentially blurry image to produce a clean, sharp version, $\mathbf{C}'_k$:

$$\mathbf{C}'_k = \Phi(\mathbf{C}_k; \mathbf{W}_{\text{NAFNet}}), \tag{5}$$

where $\mathbf{W}_{\text{NAFNet}}$ are the learned network weights. To maintain real-time performance, this computationally intensive inference executes asynchronously in a parallel thread and is strictly limited to the selected keyframes. This design ensures that the high-frequency tracking loop continues uninterrupted on the live sensor stream, while the mapping back-end receives high-fidelity, deblurred supervision. The effectiveness of this process is visually demonstrated by the 'Before/After' comparison in Fig. 3, which shows how Ro-NAFNet effectively mitigates motion blur and sensor noise. By combining these geometric and photometric filters, our pre-processing module delivers a sparse yet high-quality data stream, fundamentally reducing the introduction of uncertainty at the earliest possible stage.

C. Uncertainty-aware Gaussian Field Reconstruction

This section details our core contributions for mitigating local noise: a principled method for quantifying per-primitive uncertainty and the framework that leverages this uncertainty for intelligent, resource-efficient mapping. We begin by establishing the theoretical foundation of our uncertainty model.

*1) **Theoretical Motivation and Proxy Formulation**:* The foundation of our system is the *Uncertainty-Aware Gaussian Field*. This section establishes the theoretical role of a per-primitive confidence attribute within the broader context of SLAM state estimation, linking it to the certainty of the global map and trajectory posterior.

The central objective of SLAM is to estimate the joint posterior probability of the robot's trajectory $X_{1:t}$ and the map M given all sensor observations $Z_{1:t}$ and control inputs $U_{1:t}$:

$$P(X_{1:t}, M | Z_{1:t}, U_{1:t}). \tag{6}$$

In our framework, the map M is a dense representation composed of N 3D Gaussian primitives, $M = \{g_1, \ldots, g_N\}$, each defined by its parametric state $\boldsymbol{\theta}_i = \{\boldsymbol{\mu}_i, \mathbf{q}_i, \mathbf{s}_i, \mathbf{c}_i, \alpha_i\}$. The quality of the global SLAM estimate in Eq. (6) is fundamentally limited by the certainty of its constituent parts. A poorly-constrained or uncertain primitive introduces ambiguity that degrades the convergence and precision of the entire state estimation process.

To formalize this, we consider the marginal posterior of a single map primitive, $P(\boldsymbol{\theta}_i | Z_{1:t})$, which represents how well its parameters are constrained by the full history of observations. A formal measure of this distribution's uncertainty is

its Shannon entropy, $H(P(\boldsymbol{\theta}_i|Z_{1:t}))$. A low-entropy posterior corresponds to a well-constrained primitive with high certainty (high information), while a high-entropy posterior signifies substantial ambiguity (low information).

However, directly computing this high-dimensional posterior distribution—let alone its entropy—for millions of primitives in real-time is computationally intractable. Instead of an exact derivation, we propose a principled heuristic that serves as a differentiable proxy for this posterior uncertainty. We introduce a scalar confidence attribute, $C_i \in [0, 1]$, designed to be inversely related to the posterior entropy. A high confidence $C_i \approx 1$ approximates a low-entropy, high-information posterior, while a low confidence $C_i \approx 0$ represents a high-entropy, poorly-constrained state.

We, therefore, model confidence using observable quantities from the optimization process—namely, the gradient magnitude and opacity—that serve as strong indicators of posterior entropy. A consistently large gradient magnitude for a primitive indicates a poor fit to the observational data, suggesting that the posterior remains diffuse and high-entropy. Conversely, a converged primitive with a small gradient has found a state that well explains the data, indicative of a sharply peaked, low-entropy posterior. Similarly, a primitive's opacity (α_i) reflects the amount of observational information it has received. A near-zero opacity implies the primitive has not contributed to the rendering, leaving its posterior unconstrained and high-entropy. A high opacity, however, means its parameters have been meaningfully constrained by photometric losses, thus reducing the posterior's entropy. By combining these two indicators, our confidence metric provides a principled, real-time approximation of the underlying information content of each primitive.

*2) **Uncertainty-Driven Mapping Framework**:* Having established the theoretical role of confidence, we now detail the three core components of our practical, uncertainty-aware mapping framework: confidence estimation, keyframe selection, and adaptive optimization.

Confidence Estimation via Principled Heuristics. To estimate the confidence C_i for each primitive in real-time, a differentiable module is proposed that approximates the ideal Bayesian update by adhering to a set of first-principle inductive biases:

- Monotonicity: Confidence must be monotonically decreasing with respect to the gradient magnitude, as a large gradient signifies a poor fit to the observations.
- Boundary Conditions: In the absence of error (gradient $\to 0$), confidence should approach unity ($C_i \to 1$). If a primitive is unobserved (opacity $\alpha_i \to 0$), its confidence should revert to a neutral, high-entropy state ($C_i \to 0.5$).
- Global Adaptation: Confidence should be assessed relative to the current global state of the model, not based on fixed thresholds.

Following these principles, as shown in Fig. 4, a weighted gradient magnitude g_i is first computed for each Gaussian. Both the gradient and the opacity α_i are then normalized using global statistics (e.g., the mean over all primitives) to achieve adaptivity:

$$g_{\mathrm{n},i} = \frac{g_i}{\bar{g}}, \quad \alpha_{\mathrm{n},i} = \frac{\alpha_i}{\bar{\alpha}}. \tag{7}$$

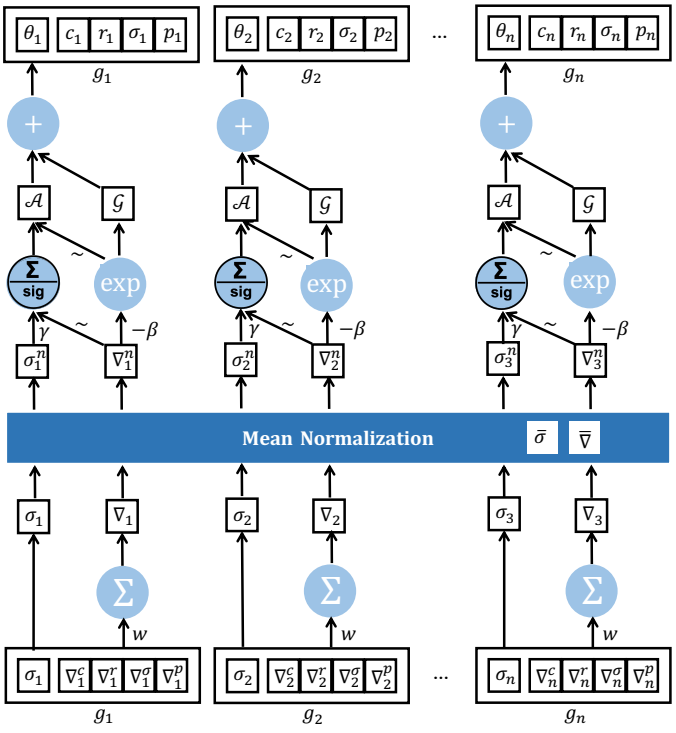

Figure 4. **Principled approximation of per-primitive confidence.** Our differentiable module approximates the ideal Bayesian update by mapping observable optimization dynamics—gradient magnitude (∇_i) and opacity (σ_i)—to a confidence value C_i. The architecture explicitly enforces our inductive biases, using mean normalization for global adaptation and a smooth gating function to combine the two input signals into a robust, real-time uncertainty estimate.

The final confidence C_i is computed via a differentiable gating mechanism that smoothly interpolates between a gradient-dominant term and an opacity-aware adjustment, satisfying all aforementioned principles:

$$\begin{aligned} C_i = \; & \exp(-\beta g_{\mathrm{n},i}) \\ & + (1 - \exp(-\beta g_{\mathrm{n},i})) \cdot \sigma\left(\gamma \alpha_{\mathrm{n},i}(1 - g_{\mathrm{n},i})\right), \end{aligned} \tag{8}$$

where β and γ are hyperparameters, and $\sigma(\cdot)$ is the sigmoid function. This formulation provides an efficient, principled approximation of per-primitive uncertainty.

Information-Theoretic Keyframe Selection. To maintain real-time performance, a SLAM system must judiciously select which frames to process for mapping. This is formulated as an active SLAM problem, where the goal is to select the next keyframe z_k that maximizes the mutual information between the map M and the new observation, thereby maximally reducing the map's posterior uncertainty. The optimal keyframe, z_k^*, is thus the one that satisfies:

$$\begin{aligned} z_k^* &= \arg\max_{z_k} I(M; z_k | Z_{1:k-1}) \\ &= \arg\max_{z_k} \left(H(M|Z_{1:k-1}) - H(M|Z_{1:k-1}, z_k)\right), \end{aligned} \tag{9}$$

where $I(\cdot;\cdot)$ is the mutual information and $H(\cdot)$ is the Shannon entropy. A direct evaluation of this expression is

Algorithm 1 Information-Theoretic Keyframe Selection

Require: Candidate frame F_k with pose $\mathbf{T}_k$, UGF Map M, Keyframe History $\mathcal{H}_{kf}$, Scale factor k_{select}

function ShouldSelectKeyframe($F_k, \mathbf{T}_k, M, \mathcal{H}_{kf}$)
▷ Applied only if F_k passes geometric filters

▷ Render the confidence map and calculate information gain
$C_{\text{map}} \leftarrow \text{RenderConfidenceMap}(M, \mathbf{T}_k)$
$\mathcal{I}(\mathbf{T}_k) \leftarrow \sum_p (1 - C_{\text{map}}[p]) \cdot M[p]$

▷ Compute the adaptive threshold
$\bar{\mathcal{I}}_{\text{recent}} \leftarrow \text{AverageGainOfRecentKeyframes}(\mathcal{H}_{kf})$
$\mathcal{I}_{\text{thresh}} \leftarrow k_{\text{select}} \cdot \bar{\mathcal{I}}_{\text{recent}}$

▷ Apply the decision rule
if $\mathcal{I}(\mathbf{T}_k) > \mathcal{I}_{\text{thresh}}$ **then**
    return `True` ▷ Frame is informative
else
    return `False` ▷ Frame is redundant
end if
end function

intractable in real-time. A key insight is that the confidence map, C_{map}, rendered from a candidate viewpoint serves as a direct and powerful proxy for the expected information gain. As established in Sec. III-C1, confidence is inversely related to entropy. Therefore, maximizing information gain is equivalent to observing regions of low confidence.

This insight is operationalized into a practical strategy where the expected information gain, $\mathcal{I}(\mathbf{T}_k)$, is approximated by integrating the uncertainty over the rendered confidence map:

$$\mathcal{I}(\mathbf{T}_k) = \sum_{p \in \text{pixels}} (1 - C_{\text{map}}[p]) \cdot M[p], \tag{10}$$

where $M[p]$ is a mask for valid regions. The decision to select a frame is based on an adaptive threshold, which is dynamically set based on the running average of recent keyframes' information gain:

$$\mathcal{I}_{\text{thresh}}(k) = k_{\text{select}} \cdot \bar{\mathcal{I}}_{\text{recent}}, \tag{11}$$

where k_{select} is a scaling factor (e.g., 0.8), and $\bar{\mathcal{I}}_{\text{recent}}$ denotes the average information gain computed over the last N_{kf} selected keyframes. This principled approach, summarized in Algorithm 1, transforms the complex problem of maximizing mutual information into a single, efficient rendering pass.

Confidence-Modulated Adaptive Optimization. Finally, the confidence C_i is leveraged to guide the optimization process itself. Interpreting C_i as inversely proportional to the variance of the gradient estimate allows for the formulation of a per-primitive adaptive learning rate:

$$\boldsymbol{\theta}_i^{t+1} = \boldsymbol{\theta}_i^t - \eta_i \nabla_{\boldsymbol{\theta}_i} \mathcal{L}, \quad \text{where} \quad \eta_i = \eta_{\text{base}} \cdot f_{\text{lr}}(C_i). \tag{12}$$

Here, $\mathcal{L}$ denotes the objective function being minimized in the current stage (e.g., the joint loss $\mathcal{L}_{\text{local}}$ detailed in Sec. III-D2). The per-primitive learning rate η_i is the core of our adaptive optimization, corresponding to the component-wise rates (e.g., η_c, η_r for color and radius) illustrated in the system architecture (Fig. 2).

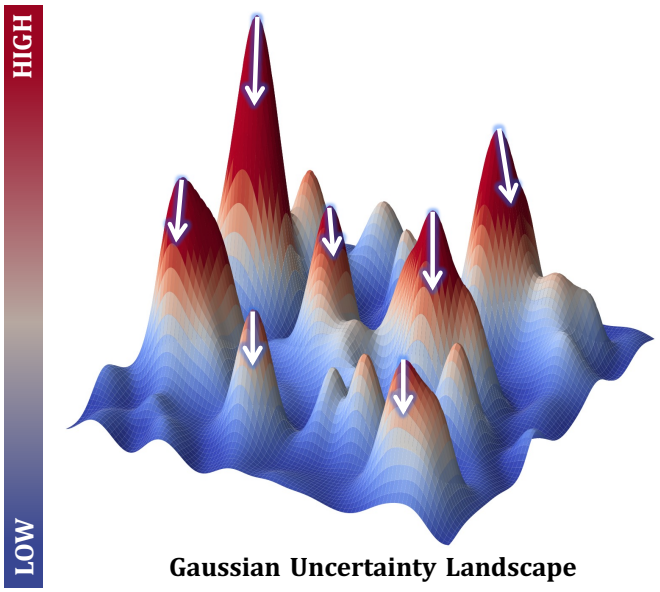

Figure 5. **Conceptual illustration of our uncertainty-driven, per-Gaussian adaptive optimization.** Each peak represents a Gaussian primitive's confidence distribution. The height corresponds to the uncertainty level, with color transitioning from blue (low uncertainty) to red (high uncertainty). The white arrows indicate the optimization step size, which is larger in high-uncertainty regions and smaller in low-uncertainty regions.

The scaling function $f_{\text{lr}}(C_i)$ is designed to follow a "cautiously aggressive" strategy (Fig. 5): high learning rates are applied to low-confidence primitives for rapid correction, while low learning rates are used for high-confidence primitives to allow for fine-tuning. This is implemented via a smooth sigmoid adjustment function:

$$f_{\text{lr}}(C_i) = f_{\min} + (f_{\max} - f_{\min}) \cdot \sigma(-s \cdot (2C_i - 1)), \tag{13}$$

where s controls the transition steepness and $[f_{\min}, f_{\max}]$ define the scaling range. This dynamic allocation of learning rates ensures that computational effort is prioritized towards refining the most uncertain parts of the map, dramatically improving convergence efficiency and robustness.

D. Tracking and Local Bundle Adjustment

This stage is responsible for the core task of maintaining high-fidelity local tracking and mapping. It serves as the primary mechanism for integrating new sensor data, robustly estimating the camera's trajectory on a frame-by-frame basis, and consistently refining the local area of the map to prevent the accumulation of errors that would require global correction. This process is divided into two key components: lightweight per-frame tracking and a more comprehensive Local Gaussian Bundle Adjustment.

1) ***Per-Frame Tracking****:* For each incoming pre-processed frame F_k, our primary goal is to robustly and efficiently estimate its camera pose $\mathbf{T}_k \in SE(3)$. The process begins by initializing the pose with a strong prior from a visual

odometry motion model. This initial estimate is then refined through a lightweight, frame-to-map optimization. Specifically, we minimize the photometric loss (Eq. (15)) between the observed image and a rendered view from the active Uncertainty-Aware Gaussian Field (UGF). To enhance robustness against uncertain map regions during tracking, this optimization is also guided by our confidence-aware principles, down-weighting the influence of low-confidence primitives. During this tracking-only step, the map primitives themselves are held constant.

*2) **Local Gaussian Bundle Adjustment (LGBA)**:* When a new keyframe F_{kf}, selected by our information-theoretic strategy (Sec. III-C2), is added to the map, we trigger a Local Gaussian Bundle Adjustment (LGBA). This is our novel mechanism for jointly optimizing the geometry and photometry of the local scene. The objective of LGBA is to refine the poses of a local window of covisible keyframes and the parameters of the UGF primitives they observe, thereby enforcing local consistency and seamlessly integrating new information.

The local optimization window is constructed around the new keyframe F_{kf}. It comprises the set of local keyframes, $\mathcal{K}_{\text{local}}$, which includes F_{kf} and its neighbors in the covisibility graph, and the set of local map primitives, $\mathcal{M}_{\text{local}}$, containing all UGF primitives (both uncertainty-aware $\mathcal{G}_U$ and shadow $\mathcal{G}_S$) observed by these keyframes.

The LGBA minimizes a joint objective function that synergistically combines photometric and geometric residuals:

$$\underset{\mathcal{K}_{\text{local}},\mathcal{M}_{\text{local}}}{\text{minimize}} \quad \mathcal{L}_{\text{local}} = \mathcal{L}_{\text{photo}} + \lambda_{\text{geo}}\mathcal{L}_{\text{geo}}. \tag{14}$$

Due to the significant difference in the scale and convergence properties of the photometric and geometric terms, a naive joint minimization can lead to unstable results. Therefore, we employ an *alternating optimization strategy*. Within a single LGBA pass, we perform several iterations optimizing only the photometric term $\mathcal{L}_{\text{photo}}$ to refine the dense structure, followed by several iterations optimizing only the geometric term $\mathcal{L}_{\text{geo}}$ to enforce rigid consistency. This decoupled approach allows each component to converge effectively.

The *photometric term*, $\mathcal{L}_{\text{photo}}$, is a weighted sum of color and depth rendering losses computed over all keyframes in $\mathcal{K}_{\text{local}}$. It is defined as:

$$\mathcal{L}_{\text{photo}} = \sum_{k\in\mathcal{K}_{\text{local}}} \left(\mathcal{L}_{\text{color}}(k) + \lambda_{\text{depth}}\mathcal{L}_{\text{depth}}(k)\right), \tag{15}$$

where the color component $\mathcal{L}_{\text{color}}(k)$ for each keyframe combines L1 and SSIM losses:

$$\mathcal{L}_{\text{color}}(k) = (1-\lambda_{\text{SSIM}})\mathcal{L}_1(\hat{\mathbf{C}}_k, \mathbf{C}_{\text{gt},k}) + \lambda_{\text{SSIM}}\mathcal{L}_{\text{SSIM}}(\hat{\mathbf{C}}_k, \mathbf{C}_{\text{gt},k}). \tag{16}$$

The depth component $\mathcal{L}_{\text{depth}}(k)$ is the L1 loss between the rendered depth map and the ground-truth depth map for the same keyframe:

$$\mathcal{L}_{\text{depth}}(k) = \mathcal{L}_1(\hat{D}_k, D_{\text{gt},k}), \tag{17}$$

where $\hat{D}_k$ and $D_{\text{gt},k}$ are the rendered and ground-truth depth maps, respectively. The optimization of this entire photometric term is critically guided by our confidence-modulated adaptive optimizer (Sec. III-C2).

The *geometric term*, $\mathcal{L}_{\text{geo}}$, enforces rigid geometric consistency using the embedded sparse feature information from the shadow Gaussians. It is formulated as the sum of reprojection errors:

$$\mathcal{L}_{\text{geo}} = \sum_{k\in\mathcal{K}_{\text{local}}} \sum_{j\in\mathcal{G}_S} \rho\left(\|\pi(\mathbf{T}_k, \boldsymbol{\mu}_j) - \mathbf{p}_{k,j}\|^2_{\Sigma}\right), \tag{18}$$

where $\pi(\cdot)$ is the camera projection function, $\mathbf{p}_{k,j}$ is the 2D observation of the shadow Gaussian's center $\boldsymbol{\mu}_j$ in keyframe k, Σ is the covariance associated with the feature scale, and $\rho(\cdot)$ is a robust Huber cost function.

The input to the LGBA is the set of covisible keyframes and the corresponding local UGF. The output is a locally consistent set of optimized keyframe poses and a refined local UGF. By jointly minimizing both photometric and geometric errors in this confidence-aware manner, our LGBA robustly integrates new observations, *enforces local consistency to prevent drift*, and maintains a high-fidelity map representation in the vicinity of the camera's current location.

E. Loop Detection and Correction for Global Consistency

While the Local Gaussian Bundle Adjustment (Sec. III-D2) effectively mitigates local errors, ensuring long-term consistency requires a robust global loop closure mechanism. Our approach is designed to eliminate long-term drift through a synergistic, *coarse-to-fine loop correction process*. This process begins with robust loop detection, followed by a high-precision relative pose estimation that leverages both the sparse geometric anchors and the dense photometric information of our unified map. The final output is a highly accurate relative pose constraint, $\mathbf{T}_{c,k} \in SE(3)$, between the current keyframe F_k and a loop candidate keyframe F_c.

*1) **Loop Candidate Detection**:* The process begins with efficient place recognition to identify potential loop closure candidates. Following established practices in sparse-feature SLAM, we utilize a DBoW2-based [14] bag-of-words approach to query a database of historical keyframes for candidates with high visual similarity. Specifically, each Shadow Gaussian retains the original ORB descriptor from its source keyframe, enabling efficient descriptor-based retrieval without re-extraction.

To reject perceptual aliasing and ensure geometric consistency, each candidate undergoes a rigorous geometric verification step. This is where our embedded shadow Gaussians ($\mathcal{G}_S$) are critical. We establish *2D-3D correspondences* by matching 2D feature observations in the current frame to the 3D centers of shadow Gaussians associated with the candidate keyframe. A candidate is accepted only if a valid relative pose can be computed with a sufficient number of inliers, ensuring the geometric viability of the loop closure.

*2) **Coarse-to-Fine Loop Correction**:* Upon successful detection of a verified loop candidate F_c, the goal is to compute an accurate relative pose $\mathbf{T}_{c,k}$ that aligns the current keyframe F_k with it.

Coarse Pose Estimation. We first establish a robust initial pose estimate, $\mathbf{T}_0 = [\mathbf{R}_0, \mathbf{t}_0]$, by solving a Perspective-n-Point

(PnP) problem. The 2D-3D correspondences are derived from matching features between F_k and F_c that correspond to the same set of shadow Gaussians. We employ a robust RANSAC-based scheme (MAGSAC++ [48]) to effectively reject outlier matches and provide a reliable, albeit coarse, initial alignment.

Fine Pose Refinement via Adaptive Warping Loss. This coarse estimate $\mathbf{T}_0$ is then passed to a fine refinement stage that minimizes a novel, photometric warping loss, directly leveraging the high-fidelity rendering capabilities of the dense UGF. This refinement is highly efficient, as it relies on the fast rendering of 3DGS and typically converges in a small number of gradient descent iterations. Given the initial pose estimate, we perform a single-pass differentiable rendering of the UGF to obtain a synthetic color image I_r and depth map D_r. For each pixel $\mathbf{u}$, we back-project its corresponding 3D point and then re-project it into the current frame F_k using the pose $\mathbf{T} = [\mathbf{R}, \mathbf{t}]$ being optimized:

$$\mathbf{u}' = \pi(\mathcal{K}(\mathbf{R}(D_r(\mathbf{u}) \cdot \mathcal{K}^{-1}\mathbf{u}) + \mathbf{t})), \tag{19}$$

where $\pi(\cdot)$ is the perspective projection function and $\mathcal{K}$ is the camera intrinsics matrix. The warping loss is then defined as the sum of photometric residuals over all valid pixels Ω:

$$\mathcal{L}_{\text{warp}}(\mathbf{T}) = \frac{1}{|\Omega|} \sum_{\mathbf{u} \in \Omega} \rho_\delta(I_r(\mathbf{u}) - I_k(\mathbf{u}')), \tag{20}$$

where I_k is the observed image of the current frame and ρ_δ is the Huber loss function.

A key limitation of standard robust losses is the reliance on a manually-tuned, fixed threshold δ. We overcome this with a *self-supervised adaptive mechanism*. At each optimization iteration t, we compute the absolute residual map $\mathbf{R}_t = |I_r(\mathbf{u}) - I_k(\mathbf{u}')|_{\mathbf{u}\in\Omega}$. The adaptive threshold δ_t is then computed using robust statistics from this distribution:

$$\delta_t = \max(\alpha \cdot \text{median}(\mathbf{R}_t), \delta_{\min}), \tag{21}$$

where α is a scaling factor (typically > 1, e.g., 1.5) that defines the boundary of the inlier region relative to the median residual, and $\delta_{\min}$ is a minimum threshold for numerical stability. This adaptive approach is theoretically grounded, providing several key advantages:

- **Robustness to Outliers:** The median is inherently robust to outliers, ensuring the threshold is not skewed by extreme residuals.
- **Scale Adaptation:** The threshold automatically adapts to the current magnitude of the residuals, allowing for aggressive L1-like updates in early stages and fine-grained L2-like quadratic refinement as optimization converges.
- **Convergence Stability:** This dynamic adjustment prevents both over-smoothing of details and over-sensitivity to noise, leading to more stable and accurate convergence.

The resulting optimized pose $\mathbf{T}_{c,k}$ provides a high-precision constraint that is sent to the final global optimization stage.

F. Global Optimization and Map Correction

The final stage of our framework is responsible for ensuring *true global consistency* by propagating the high-precision loop closure constraint throughout the entire map and trajectory. Directly optimizing the millions of Gaussian primitives and all keyframe poses simultaneously is computationally prohibitive. Therefore, we design a principled, two-stage global optimization process that first corrects the sparse geometric backbone and then efficiently brings the dense, photometric map into alignment.

*1) **Stage 1 - Pose Graph Optimization**:* The first stage addresses the global trajectory drift. We construct an *essential graph* [13] where nodes represent all keyframe poses $\{\mathbf{T}_i\}$ and edges represent geometric constraints from both covisibility and the new loop closure. We then perform a *Pose Graph Optimization (PGO)* to efficiently solve for the optimal set of corrected keyframe poses, $\{\mathbf{T}_i^*\}$:

$$\{\mathbf{T}_i^*\} = \underset{\{\mathbf{T}_i\}}{\text{argmin}} \Bigg(\sum_{(i,j)\in\mathcal{E}_{\text{covis}}} \| \log((\mathbf{T}_i^{-1}\mathbf{T}_j)\mathbf{T}_{i,j}^{-1}) \|^2_{\Sigma_{i,j}} + \sum_{(c,k)\in\mathcal{E}_{\text{loop}}} \| \log((\mathbf{T}_c^{-1}\mathbf{T}_k)\mathbf{T}_{c,k}^{-1}) \|^2_{\Sigma_{c,k}} \Bigg). \tag{22}$$

where the $\log(\cdot)$ map is the $SE(3) \to \mathfrak{se}(3)$ logarithm. This sparse optimization yields a globally consistent camera trajectory $\{\mathbf{T}_i^*\}$. However, the dense UGF map, constructed based on the original, uncorrected poses $\{\mathbf{T}_i\}$, is now misaligned with this new trajectory.

*2) **Stage 2 - Global Map Correction and Refinement**:* After the PGO has established a globally consistent trajectory, the dense UGF map must be brought into alignment. Directly optimizing the millions of Gaussian primitives alongside the corrected poses would be computationally prohibitive and susceptible to poor local minima, especially in large-scale scenes with significant drift. We therefore adopt a two-stage strategy to ensure both efficiency and robustness. This is achieved in two steps: a rapid global deformation followed by a final refinement.

Global Deformation. First, we perform a computationally efficient global deformation to bring the entire map into coarse alignment with the corrected trajectory. For each Gaussian primitive g_i (both uncertainty-aware and shadow), which is associated with a reference keyframe k, we compute the corrective transformation $\Delta\mathbf{T}_k = \mathbf{T}_k^*\mathbf{T}_k^{-1}$. We then apply this transformation to the primitive's position $\boldsymbol{\mu}_i$:

$$\boldsymbol{\mu}_i^* = \Delta\mathbf{T}_k\boldsymbol{\mu}_i. \tag{23}$$

This process rigidly transforms the entire dense and sparse structure of the map, efficiently resolving the primary global inconsistency.

Global Gaussian Bundle Adjustment (GGBA). While the deformation corrects the global structure, it does not resolve any non-rigid distortions or photometric inconsistencies. To address this, we perform a final refinement via our novel *Global Gaussian Bundle Adjustment (GGBA)*. Conceptually, GGBA is a large-scale version of our LGBA (Sec. III-D2). It minimizes a joint objective function $\mathcal{L}_{\text{global}} = \mathcal{L}_{\text{photo-global}} + \lambda_{\text{geo-global}}\mathcal{L}_{\text{geo-global}}$ using an alternating optimization strategy:

- **Global Geometric BA:** This step refines the corrected keyframe poses $\{\mathbf{T}_i^*\}$ and the deformed positions of

all shadow Gaussians $\{\boldsymbol{\mu}_j^*\}$ by minimizing the global geometric reprojection error.

- **Global Photometric Refinement:** This step fine-tunes the parameters of the *uncertainty-aware Gaussians* $\{\boldsymbol{\theta}_i^U\}$, keeping the poses fixed. It is critically guided by our *confidence-modulated adaptive optimizer*.

By first deforming the map and then performing this alternating refinement, our GGBA effectively and efficiently propagates the global correction to the dense photometric representation, ensuring the final output is a globally consistent, high-fidelity Uncertainty-Aware Gaussian Field.

IV. Experiments and Discussions

This paper introduced μSLAM, an uncertainty-driven 3DGS-based SLAM system designed to achieve robust, real-time localization and photorealistic reconstruction in challenging real-world robotic scenarios. We now present a comprehensive set of experiments designed to validate the performance and capabilities of our proposed system. The results systematically support our three key claims:

1) **Superior Tracking Accuracy:** μSLAM achieves state-of-the-art tracking accuracy, outperforming existing rendering-based SLAM approaches.
2) **High-Fidelity Reconstruction:** Our approach delivers high-fidelity reconstruction quality that is on par with, or superior to, competing methods, particularly in challenging real-world scenes.
3) **Robustness with Real-Time Efficiency:** Compared to other radiance field-based methods, irrespective of their loop closure capabilities, μSLAM exhibits superior runtime efficiency and robustness. It consistently operates at over 30 FPS and successfully completes challenging sequences where competing systems often fail.

A. Datasets

We evaluate our system on a diverse set of public and self-recorded datasets to ensure a comprehensive and rigorous analysis.

Our primary evaluation is conducted on the widely-used TUM-RGBD [49] and ScanNet [50] benchmarks. From TUM-RGBD, we select a range of sequences to target specific system capabilities: those with subtle motion (e.g., `fr2/desk`) evaluate the efficiency of our keyframe selection, while those with aggressive motion (e.g., `fr1/360`) test robustness against severe motion blur. To assess long-term performance and generalization, we utilize the large-scale sequences from TUM-RGBD (e.g., `fr2/pioneer-slam`) and the ScanNet dataset, both of which feature complex, noisy, real-world indoor environments.

To further validate performance in authentic robotic operation scenarios, we also collected several long-term RGB-D sequences using a Dashgo-D1 mobile robot. These sequences were captured in challenging indoor environments and contain realistic sensor noise and motion artifacts common to wheeled robotic platforms.

Table I
KEY HYPERPARAMETERS OF OUR μSLAM SYSTEM.

Parameter	Symbol	Value
Pre-processing Filters (Sec. III-B)		
Translational Baseline Threshold	τ_{trans}	$0.15m$
Tracking Continuity Threshold	τ_{track}	15
View Overlap Threshold	τ_{overlap}	0.7
Uncertainty-Aware Framework (Sec. III-C)		
Confidence Parameter (Gradient)	β	100.0
Confidence Parameter (Opacity)	γ	10.0
Adaptive LR Min Factor	f_{min}	0.1
Adaptive LR Max Factor	f_{max}	2.0
Optimization Parameters (Sec. III-D)		
Tracking Iterations per Frame	N_{track}	200
Mapping Iterations per KF	N_{map}	20
SSIM Loss Weight	λ_{SSIM}	0.3
Base Opacity LR	$\eta^{\alpha}_{\text{base}}$	0.15
Base Position LR	η^{μ}_{base}	0.0003
Base Color LR	η^{c}_{base}	0.009
Loop Closure (Sec. III-E)		
DBoW2 Similarity Threshold	τ_{DBoW2}	0.75
Warping Loss Huber Parameter	α	1.5

B. Implementation Details

We summarize the key hyperparameters of our system in Tab. I. These settings are kept consistent across all experiments to ensure a fair and reproducible evaluation. Our system is implemented in PyTorch and all experiments are conducted on a workstation with an NVIDIA RTX 4090 GPU. The visual odometry front-end is a modified version of ORB-SLAM3, and the pose graph optimization is performed using g2o [51].

Most hyperparameters, detailed in Tab. I, are kept fixed across all experiments. For instance, the adaptive learning rate bounds, f_{min} and f_{max}, are set to 0.1 and 2.0 respectively. These values were determined through initial experiments to effectively implement our "cautiously aggressive" optimization strategy: they ensure that high-confidence primitives are fine-tuned with a significantly reduced learning rate (down to 10% of the base rate), while allowing low-confidence primitives to be corrected rapidly with a doubled rate. This strategy proved robust across all tested scenarios. The primary exception is the keyframe selection scaling factor, k_{select} (Eq. (11)), which we adjust on a per-dataset basis. This is necessary to account for different camera frame rates and typical motion speeds, ensuring an appropriate density of keyframes for each specific dataset.

C. Tracking Performance (Claim i)

The first set of experiments is designed to rigorously evaluate the camera pose estimation accuracy of our system and validate our first claim. We evaluate tracking performance on the TUM-RGBD and ScanNet datasets using the *ATE RMSE* [*cm*] as the primary metric.

*1) **Performance on TUM-RGBD**:* On the TUM-RGBD dataset (Tab. II-A), our method, μSLAM, demonstrates state-of-the-art performance among all rendering-based and hybrid approaches, achieving the best average accuracy of 2.24 *cm*. This result is particularly significant when analyzing performance on sequences with distinct challenges. On sequences

with significant motion blur and texture variation like `desk2`, our method's *ATE RMSE* of 2.45 *cm* is superior to all others. This can be directly attributed to our pre-processing pipeline, which provides a cleaner supervisory signal, and our uncertainty-aware optimizer (Go-Dial), which down-weights the influence of corrupted data during optimization. Furthermore, in the large-scale `room` sequence, where long-term drift becomes a major factor, our method achieves the best result (5.35 *cm*), showcasing the effectiveness of our synergistic coarse-to-fine loop closure in eliminating accumulated error.

When compared to classical methods, μSLAM significantly outperforms dense fusion approaches like Kintinuous and ElasticFusion. While the highly-optimized sparse method ORB-SLAM2 achieves a slightly better average *ATE RMSE*, it does so at the cost of providing only a sparse map. Our approach provides a dense, photorealistic reconstruction while maintaining a tracking accuracy that is highly competitive with the best sparse methods, striking a superior balance between localization precision and mapping fidelity.

2) ***Performance on ScanNet****:* The ScanNet dataset poses additional challenges with its larger scale, more complex geometry, and significant sensor noise, making a robust loop closure mechanism critical for success. As shown in Tab. II-B, SLAM methods that lack an explicit and robust loop closure mechanism (e.g., NICE-SLAM, SplaTAM, MonoGS) suffer from substantial pose estimation errors, highlighting the difficulty of the task.

In this challenging benchmark, our method achieves an average *ATE* of 6.84 *cm*, ranking second and performing on par with the top method, Go-SLAM. This strong performance validates the robustness and generalization capability of our loop closure strategy. Our ability to first secure a robust coarse alignment using the embedded sparse shadow Gaussians and then refine it with the dense map's photometric information allows our system to successfully close loops even in visually complex and noisy environments. It is important to note that the leading method, Go-SLAM, is a non-real-time system that relies on a heavyweight front-end. In contrast, μSLAM achieves this competitive accuracy while operating entirely in real-time, making it far more suitable for practical robotic applications. This result underscores our system's ability to achieve state-of-the-art accuracy without compromising on the critical requirement of real-time efficiency.

D. Reconstruction Quality and Global Consistency (Claim ii)

This set of experiments evaluates the reconstruction quality, which we also use as a quantitative proxy for underlying map consistency. The results support our second claim that μSLAM enables high-fidelity, globally consistent reconstruction suitable for online robotic applications. We evaluate rendering quality on novel views using *PSNR, SSIM* [55], and *LPIPS* [56], with the reasoning that significant geometric inconsistencies will manifest as rendering artifacts that degrade these photometric scores.

1) ***Performance on TUM-RGBD****:* On the TUM-RGBD dataset (Tab. III-A), our method achieves the best average reconstruction quality among all evaluated systems, with a *PSNR* of 23.20 and *LPIPS* of 0.15. This is a substantial improvement over other real-time hybrid systems like GS-ICP-SLAM (20.59 *PSNR*) and demonstrates a superior balance of speed and fidelity. Furthermore, our clear advantage over non-loop-closure methods like SplaTAM in both tracking (Sec. IV-C) and mapping quality highlights the critical role of our global consistency mechanism.

2) ***Analysis of Global Consistency on ScanNet****:* The importance of global consistency becomes paramount on the larger ScanNet dataset (Tab. III-B). Here, a direct comparison of final rendering metrics can be misleading. For instance, LoopSplat achieves the highest scores, but only after an extensive offline post-processing pipeline. To create a fair comparison of *online* map quality, we compare against LoopSplat*, a variant using a simple merge of its online submaps. The advantage of our globally consistent online map is evident: our method's average *PSNR* of 21.87 is dramatically higher than that of LoopSplat* (12.04 *PSNR*), whose low score reflects the severe visual artifacts from submap misalignment.

Our primary competitor for globally consistent *online* reconstruction is 2DGS-SLAM. While its average *PSNR* (21.57) is competitive with ours (21.87), our method demonstrates a clear advantage in perceptual quality, achieving a significantly better average *LPIPS* score (0.30 vs. 0.43 for 2DGS-SLAM). This indicates that our reconstructions are not only geometrically consistent but also more photorealistic.

In summary, the quantitative results validate that μSLAM produces a more consistent and higher-fidelity online map than competing approaches. It surpasses other real-time methods in quality and delivers superior perceptual fidelity compared to previous non-real-time state-of-the-art systems, all without reliance on heavy offline post-processing.

E. Robustness and Real-time Performance (claim iii)

The final set of experiments evaluates the robustness and real-time efficiency of μSLAM, providing quantitative support for our third claim. We demonstrate that our approach achieves a superior balance between these two critical metrics compared to other radiance field-based methods, particularly in challenging robotic scenarios involving high view overlap, aggressive motion, and long-term navigation. We assess performance using several key indicators. System efficiency is quantified by the average Frames Per Second (*FPS*). Robustness is measured primarily by the success rate across 13 challenging sequences from the TUM-RGBD dataset, where success is defined as the completion of a sequence without tracking failure. To further quantify the final output quality under these stressful conditions, we also report the tracking accuracy (*ATE RMSE*) and reconstruction quality (*PSNR*, *SSIM*, *LPIPS*) for all successfully completed runs. For comparison, we select state-of-the-art baselines from the previous experiments, including both hybrid (Photo-SLAM [44], GS-ICP-SLAM [43]) and non-hybrid (MonoGS [35], SplaTAM [37]) systems.

1) ***Quantitative Analysis****:* As reported in TABLE V, μSLAM achieves a 100% success rate across all 9 challenging sequences. This matches the robustness of the non-real-time Photo-SLAM and exceeds that of SplaTAM (67% success)

Table II
ABSOLUTE TRAJECTORY ERROR (ATE) ON THE TUM-RGBD AND SCANNET DATASETS, REPORTED IN CENTIMETERS. **LC**: LOOP CLOSURE; **HY**: HYBRID (SPARSE+DENSE); **RT**: REAL-TIME. BEST RESULTS IN EACH CATEGORY ARE IN **BOLD**, SECOND BEST ARE UNDERSCORED.

(A) TUM-RGBD Dataset

Method	LC	HY	RT	desk	desk2	room	xyz	office	Avg.
Rendering-based Approaches									
NICE-SLAM [27]	✗	✗	✗	4.26	4.99	34.49	6.19	3.87	10.76
E-SLAM [30]	✗	✗	✗	2.47	3.69	29.73	1.11	2.42	7.89
Point-SLAM [32]	✗	✗	✗	4.34	4.54	30.92	1.31	3.48	8.92
Loopy-SLAM [33]	✓	✗	✗	3.79	3.38	7.03	1.62	3.41	3.85
MonoGS [35]	✗	✗	✗	**1.59**	7.03	8.55	1.44	1.49	4.02
SplaTAM [37]	✗	✗	✗	3.35	6.54	11.13	1.24	5.16	5.48
Gaussian-SLAM [38]	✗	✗	✗	2.73	6.03	14.92	1.39	5.31	6.08
Photo-SLAM [44]	✗	✓	✗	1.87	2.86	5.77	0.38	**1.42**	2.46
GS-ICP-SLAM [43]	✗	✓	✓	2.82	6.7	10.59	1.77	2.94	4.96
LoopSplat [45]	✓	✗	✗	2.08	3.54	6.24	1.58	3.22	3.33
2DGS-SLAM [12]	✓	✗	✗	1.84	2.76	5.98	1.16	1.97	2.74
μSLAM (Ours)	✓	✓	✓	1.67	**2.45**	**5.35**	**0.30**	**1.42**	**2.24**
Classical SLAM Approaches									
Kintinuous [18]	✓	✗	✓	3.7	7.1	7.5	2.9	3.0	4.84
ElasticFusion [25]	✓	✗	✓	2.0	4.8	6.8	1.1	1.7	3.28
ORB-SLAM2 [52]	✓	✗	✓	**1.6**	**2.2**	**4.7**	**0.4**	**1.0**	**2.0**
RTAB-Map [53]	✓	✗	✓	2.9	4.4	6.6	0.5	2.1	3.3

(B) ScanNet Dataset

Method	LC	HY	RT	00	59	106	181	207	Avg.
Rendering-based Approaches									
NICE-SLAM [27]	✗	✗	✗	12.0	14.0	7.9	13.4	6.2	10.70
Go-SLAM [54]	✗	✗	✗	5.4	7.5	**7.0**	**6.8**	6.9	**6.72**
E-SLAM [30]	✗	✗	✗	7.3	8.5	7.5	9.0	**5.7**	7.60
Point-SLAM [32]	✗	✗	✗	10.2	7.8	8.7	14.8	9.5	10.20
Loopy-SLAM [33]	✓	✗	✗	**4.2**	7.5	8.3	10.6	7.9	7.70
MonoGS [35]	✗	✗	✗	9.8	32.1	8.9	21.8	7.9	16.10
SplaTAM [37]	✗	✗	✗	12.8	10.1	17.7	11.1	7.5	11.84
Gaussian-SLAM [38]	✗	✗	✗	21.2	12.8	13.5	21.0	14.3	16.56
Photo-SLAM [44]	✗	✓	✗	–	–	–	–	–	–
GS-ICP-SLAM [43]	✗	✓	✓	–	–	–	–	–	–
LoopSplat [45]	✓	✗	✗	6.2	7.1	7.4	8.5	6.6	7.16
2DGS-SLAM [12]	✓	✗	✗	6.6	6.9	7.1	8.2	6.0	6.96
μSLAM (Ours)	✓	✓	✓	6.2	**6.4**	7.5	7.9	6.2	6.84

and GS-ICP-SLAM (89% success). The failures of competing methods occur primarily in sequences with aggressive motion or long-term trajectories, where unhandled noise or drift leads to tracking loss. This directly highlights the importance of our robust pre-processing pipeline in handling severe sensor noise and our synergistic loop closure mechanism in eliminating accumulated drift.

In terms of efficiency, μSLAM operates at a consistent average of 30 *FPS*. To provide a transparent view of our computational cost, we detail the runtime breakdown of each core module in TABLE IV. The main tracking thread operates at $\sim$30 Hz (approx. 31.2 *ms* per cycle), dedicating 21.2 *ms* to visual odometry and 10.0 *ms* to geometric keyframe pre-filtering. Crucially, the computationally intensive tasks—including the rendering-based *Shannon-MI Keyframe Selection*, *Ro-NAFNet* inference, and *Local G-BA*—are offloaded to a parallel mapping thread. These modules execute asynchronously and only on the sparse set of geometrically selected keyframes. This decoupled architecture prevents blocking the main tracking loop, allowing our system to maintain real-time performance alongside GS-ICP-SLAM, while significantly outperforming high-fidelity methods such as SplaTAM (< 1 *FPS*) and Photo-SLAM ($\sim$3 *FPS*).

Crucially, this real-time robustness does not compromise tracking accuracy or reconstruction quality. On the aggressive motion sequences, for example, μSLAM achieves both the lowest average *ATE RMSE* of 6.91 *cm* and the highest average *PSNR* of 23.52. This represents a substantial improvement in both accuracy and quality over the other real-time system, GS-ICP-SLAM (61.93 *cm*, 18.06 *PSNR*), and also surpasses the robust, non-real-time Photo-SLAM (8.10 *cm*, 22.71 *PSNR*).

*2) **Qualitative Analysis**:* The qualitative results, presented in Fig. 6, provide a visual summary of our system's performance on the TUM-RGBD benchmark. The figure showcases the online reconstruction and tracking outputs across sequences with varying motion profiles, highlighting the high fidelity and geometric consistency of the maps produced by μSLAM in real-time. The novel view synthesis results

Table III
QUANTITATIVE COMPARISON OF RECONSTRUCTION AND RENDERING QUALITY ON THE TUM-RGBD AND SCANNET DATASETS. WE REPORT PSNR, SSIM, AND LPIPS. BEST RESULTS ARE IN **BOLD**, SECOND BEST ARE UNDERSCORED.

(A) TUM-RGBD Dataset										
Method	**LC**	**HY**	**RT**	**Metrics**	desk	desk2	room	xyz	office	**Avg.**
SplaTAM [37]	✗	✗	✗	**PSNR**	22.00	–	–	24.50	21.90	22.80
				SSIM	0.86	–	–	**0.95**	**0.88**	**0.90**
				LPIPS	0.23	–	–	0.10	0.20	0.18
Photo-SLAM [44]	✗	✓	✗	**PSNR**	20.87	–	–	22.09	**22.74**	21.90
				SSIM	0.74	–	–	0.77	0.78	0.76
				LPIPS	0.24	–	–	0.17	**0.15**	0.19
GS-ICP-SLAM [43]	✗	✓	✓	**PSNR**	17.77	–	–	23.34	20.65	20.59
				SSIM	0.71	–	–	0.83	0.76	0.77
				LPIPS	0.30	–	–	0.14	0.23	0.22
LoopSplat [45]	✓	✗	✗	**PSNR**	–	–	–	–	–	22.72
				SSIM	–	–	–	–	–	0.87
				LPIPS	–	–	–	–	–	0.26
μSLAM (Ours)	✓	✓	✓	**PSNR**	**22.45**	–	–	**24.62**	22.54	**23.20**
				SSIM	**0.88**	–	–	0.91	0.85	0.88
				LPIPS	**0.19**	–	–	**0.09**	0.17	**0.15**
(B) ScanNet Dataset										
Method	**LC**	**HY**	**RT**	**Metrics**	00	59	106	181	207	**Avg.**
NICE-SLAM [27]	✗	✗	✗	**PSNR**	18.71	16.55	17.29	15.56	18.38	17.30
				SSIM	0.64	0.61	0.65	0.56	0.65	0.62
				LPIPS	0.56	0.53	0.51	0.60	0.55	0.55
Point-SLAM [32]	✗	✗	✗	**PSNR**	19.06	16.38	18.46	16.75	19.66	18.06
				SSIM	0.66	0.62	0.75	0.67	0.70	0.68
				LPIPS	0.52	0.53	0.44	0.53	0.50	0.50
MonoGS [35]	✗	✗	✗	**PSNR**	21.13	19.70	21.35	22.02	20.95	21.03
				SSIM	0.72	0.72	0.81	0.81	0.73	0.76
				LPIPS	0.45	0.44	0.34	0.43	0.46	0.42
SplaTAM [37]	✗	✗	✗	**PSNR**	19.33	19.27	17.73	16.76	19.80	18.58
				SSIM	0.66	0.79	0.69	0.68	0.70	0.70
				LPIPS	0.44	0.29	0.38	0.42	0.34	0.37
GS-ICP-SLAM [43]	✗	✓	✓	**PSNR**	✗	19.63	20.30	22.21	20.53	20.67
				SSIM	✗	0.79	0.77	0.74	0.75	0.76
				LPIPS	✗	0.32	**0.23**	0.28	**0.27**	**0.28**
LoopSplat [45]	✓	✗	✗	**PSNR**	**24.99**	**23.23**	**23.35**	**24.82**	**26.33**	**24.54**
				SSIM	**0.84**	**0.83**	**0.85**	**0.82**	**0.85**	**0.84**
				LPIPS	0.45	0.40	0.41	0.51	0.43	0.44
LoopSplat* [45]	✓	✗	✗	**PSNR**	12.35	12.95	10.26	11.47	13.17	12.04
				SSIM	0.41	0.41	0.32	0.54	0.50	0.44
				LPIPS	0.84	0.72	0.80	0.70	0.70	0.75
2DGS-SLAM [12]	✓	✗	✗	**PSNR**	23.36	19.00	20.53	21.27	23.71	21.57
				SSIM	0.77	0.73	0.80	**0.82**	0.78	0.78
				LPIPS	0.44	0.44	0.36	0.49	0.43	0.43
μSLAM (Ours)	✓	✓	✓	**PSNR**	21.62	20.46	21.86	22.62	22.79	21.87
				SSIM	0.76	0.80	0.79	0.79	0.82	0.79
				LPIPS	**0.41**	**0.26**	0.25	**0.26**	0.31	0.30

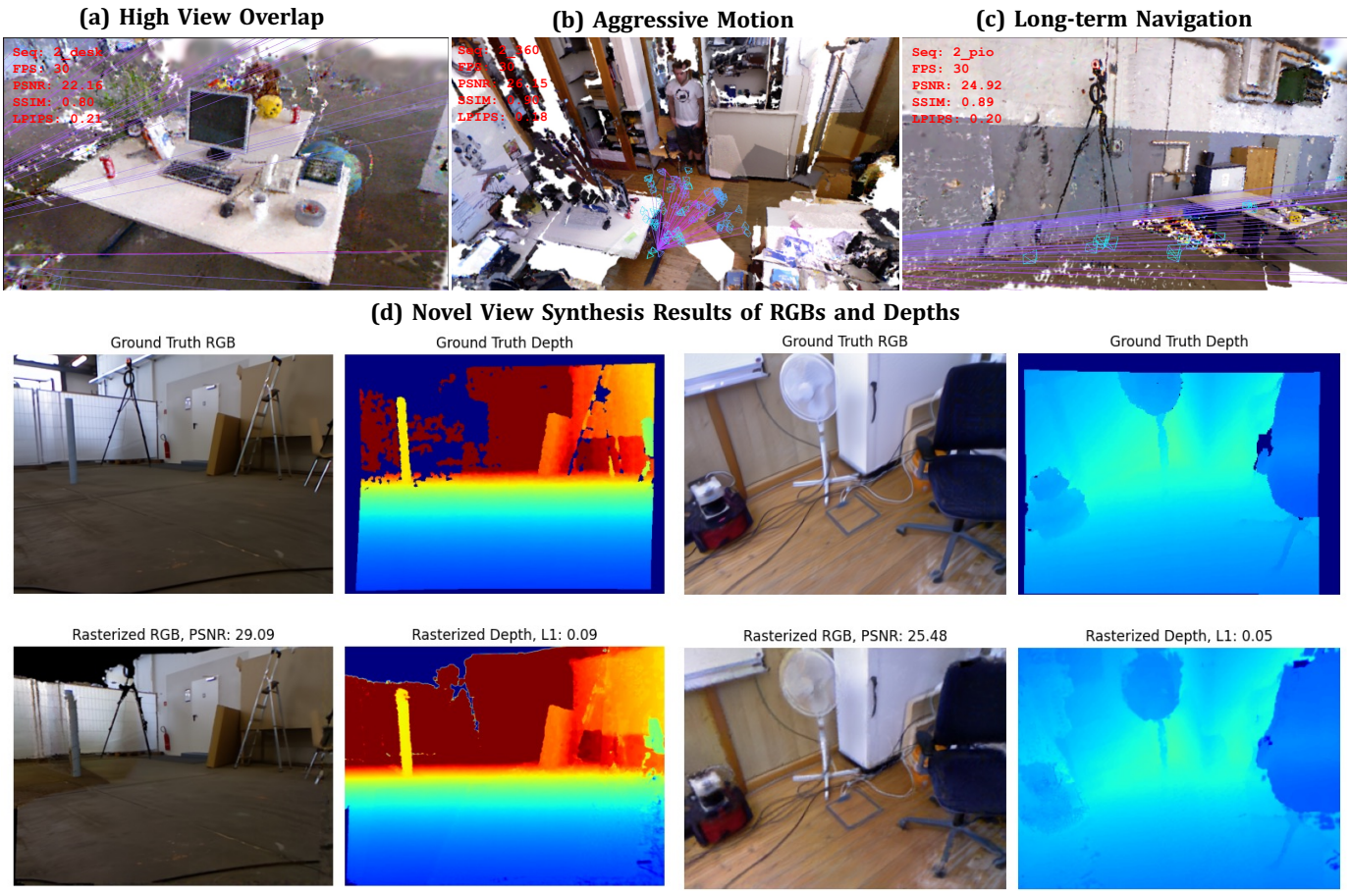

Figure 6. **Qualitative performance summary on the TUM-RGBD dataset.** This figure showcases the online output of μSLAM across three representative and challenging sequence types. **(a) High View Overlap (fr2/desk):** Our system maintains high accuracy and map quality while efficiently selecting keyframes, indicated by the tracked feature lines. **(b) Aggressive Motion (fr1/360):** Even with severe rotational motion blur, our system robustly tracks the camera (trajectory shown) and reconstructs a coherent scene. **(c) Long-term Navigation (fr2/pioneer_slam):** Our synergistic loop closure mechanism successfully corrects for long-term drift, resulting in a globally consistent map. **(d) Novel View Synthesis:** We compare rendered RGB and Depth images from our final map against the ground truth for two novel viewpoints. The high PSNR and low L1 depth error demonstrate the high fidelity and geometric accuracy of our reconstruction.

Table IV
RUNTIME BREAKDOWN PER MODULE. THE SYSTEM MAINTAINS REAL-TIME PERFORMANCE (∼30 HZ) BY EXECUTING HEAVY RENDERING AND OPTIMIZATION TASKS ASYNCHRONOUSLY ON KEYFRAMES ONLY.

Module	Time (ms)	Frequency	Thread
Tracking (Vision)	21.2	Per Frame	Main
Geometric KF Selection	10.0	Per Frame	Main
Total Tracking Loop	**∼31.2**	**∼30 Hz**	**Main**
Shannon-MI KF Selection	12.1	Keyframe Only	Parallel
Ro-NAFNet Inference	45.0	Keyframe Only	Parallel
Local G-BA	60.0	Keyframe Only	Parallel

(Fig. 6d) further confirm the high quality of both the rendered RGB and depth, demonstrating the effectiveness of our complete uncertainty-aware pipeline.

Qualitative results under severe motion blur and sensor noise are presented in Fig. 7. The comparison demonstrates that our method produces significantly sharper and more photorealistic reconstructions than other state-of-the-art rendering-based approaches. This superior performance is a direct result of our comprehensive approach to handling error: our multi-stage noise mitigation framework (pre-processing and uncertainty-aware optimization) handles per-frame artifacts, while our loop closure mechanism ensures global map consistency, preventing the large-scale distortion evident in other methods.

In summary, the results show that μSLAM is the only system evaluated that simultaneously delivers a 100% success rate, real-time performance (>30 *FPS*), state-of-the-art tracking accuracy, and superior reconstruction quality. This confirms that our uncertainty-aware architecture and integrated loop closure mechanism effectively address the trade-offs between speed, robustness, and fidelity.

F. Ablation Study

To quantitatively deconstruct the contributions of our core components, we conduct a comprehensive ablation study on challenging sequences from the TUM-RGBD dataset. We establish several system variants by incrementally adding our proposed modules to a baseline (Base), which includes only geometric pre-filtering and constraints. The variants include

Table V
QUANTITATIVE EVALUATION ON CHALLENGING REAL-WORLD SEQUENCES FROM THE TUM-RGBD DATASET, CATEGORIZED BY MOTION PROFILE. WE REPORT SUCCESS RATE (SR), EFFICIENCY (FPS), TRACKING ACCURACY (ATE RMSE), AND RECONSTRUCTION QUALITY (PSNR, SSIM, LPIPS). BEST RESULTS IN EACH COLUMN ARE IN **BOLD**, SECOND BEST ARE UNDERSCORED.

*(A) Sequences with **High View Overlap***								
Method	**SR**	**FPS**	**RMSE**	**Metrics**	`fr1-xyz`	`fr1-rpy`	`fr2-rpy`	**Avg.**
SplaTAM [37]	67%	< 1	4.21	**PSNR**	22.89	✗	21.90	22.40
				SSIM	**0.91**	✗	**0.86**	0.89
				LPIPS	**0.15**	✗	0.22	**0.185**
Photo-SLAM [44]	**100%**	∼3	1.07	**PSNR**	**23.83**	20.71	**23.21**	22.58
				SSIM	0.84	0.76	0.77	0.79
				LPIPS	–	–	–	–
GS-ICP-SLAM [43]	**100%**	**>30**	2.33	**PSNR**	19.61	17.83	22.06	19.83
				SSIM	0.74	0.72	0.82	0.76
				LPIPS	0.52	0.30	**0.17**	0.33
μSLAM (Ours)	**100%**	**>30**	**1.01**	**PSNR**	22.94	**23.58**	22.17	**22.90**
				SSIM	0.89	**0.93**	**0.86**	**0.89**
				LPIPS	0.17	**0.19**	0.21	0.19
*(B) Sequences with **Aggressive Motion***								
Method	**SR**	**FPS**	**RMSE**	**Metrics**	`1-360`	`1-floor`	`2-hemi`	**Avg.**
SplaTAM [37]	67%	< 1	62.43	**PSNR**	18.41	18.77	✗	18.81
				SSIM	0.74	0.65	✗	0.72
				LPIPS	0.27	0.38	✗	0.33
Photo-SLAM [44]	**100%**	∼3	8.10	**PSNR**	23.74	**23.73**	21.28	22.71
				SSIM	**0.75**	**0.75**	0.75	0.75
				LPIPS	–	–	–	–
GS-ICP-SLAM [43]	67%	**>30**	61.93	**PSNR**	17.01	18.44	✗	18.06
				SSIM	0.71	0.65	✗	0.68
				LPIPS	0.36	0.46	✗	0.39
μSLAM (Ours)	**100%**	**>30**	**6.91**	**PSNR**	**26.15**	23.65	**22.63**	**23.52**
				SSIM	**0.90**	**0.75**	**0.77**	**0.82**
				LPIPS	**0.18**	**0.24**	**0.24**	**0.20**
*(C) Sequences with **Long-term Navigation***								
Method	**SR**	**FPS**	**RMSE**	**Metrics**	`2-nloop`	`2-wloop`	`2-pio`	**Avg.**
SplaTAM [37]	67%	< 1	286.71	**PSNR**	24.33	✗	13.62	19.26
				SSIM	0.82	✗	0.42	0.69
				LPIPS	**0.22**	✗	0.54	0.31
Photo-SLAM [44]	**100%**	∼3	9.09	**PSNR**	20.17	19.25	21.16	19.33
				SSIM	0.76	0.74	0.79	0.73
				LPIPS	–	–	–	–
GS-ICP-SLAM [43]	**100%**	**>30**	214.26	**PSNR**	21.52	21.62	21.37	21.50
				SSIM	0.81	**0.84**	0.79	0.81
				LPIPS	0.32	0.31	0.34	0.32
μSLAM (Ours)	**100%**	**>30**	**8.94**	**PSNR**	**25.13**	**23.54**	**24.92**	**23.13**
				SSIM	**0.83**	0.83	**0.89**	**0.81**
				LPIPS	0.29	**0.24**	**0.20**	**0.24**

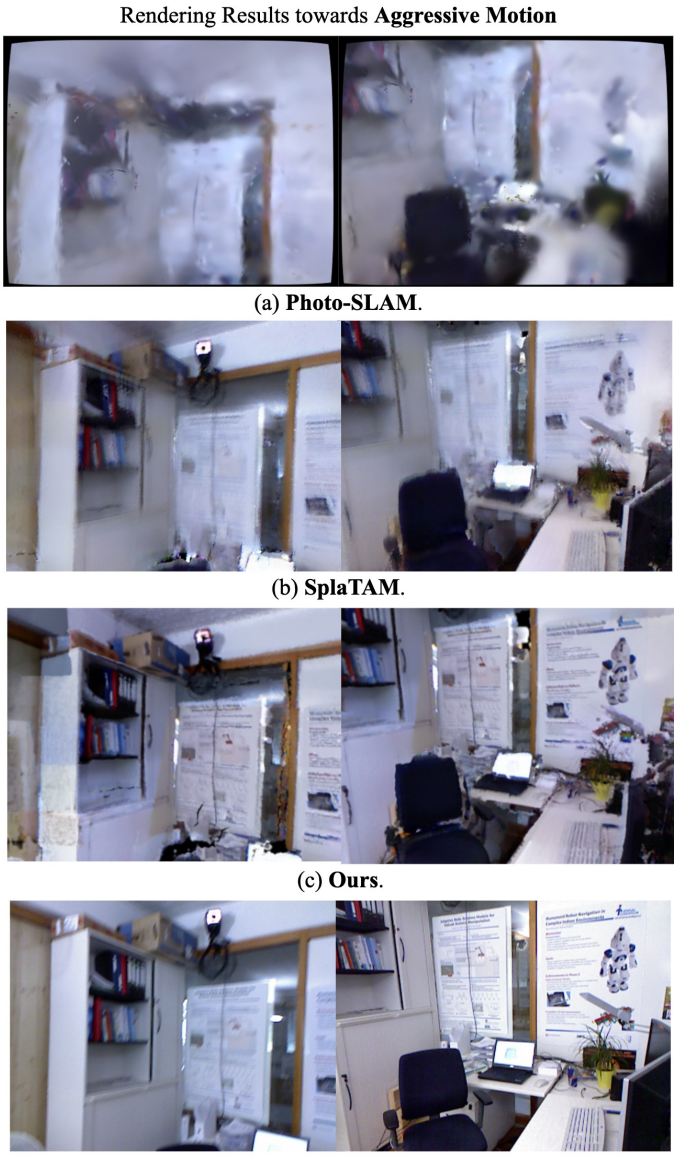

Figure 7. **Robustness to Motion Blur and Sensor Noise.** A qualitative comparison on a challenging real-world sequence (1-360) with aggressive motion at the loop closure. Our method (c) successfully reconstructs fine details. In contrast, competing methods suffer from noticeable artifacts, such as ghosting from trajectory drift (a) and blurring from unhandled motion artifacts (b), demonstrating the superiority of our uncertainty-aware framework.

adding our adaptive optimizer (+Go-Dial), our information-theoretic keyframe selector (+Go-View), and the full system (Full).

The results, summarized in Tab. VI, demonstrate the critical role and synergistic interplay of each module.

1) ***Contribution of Individual Modules****:* The Base system, while functional, suffers from two key limitations: inefficient optimization due to a uniform learning rate, and a high computational load from processing all frames, preventing real-time performance ($<$ 10 *FPS*).

The introduction of the Go-Dial module directly addresses optimization inefficiency. By dynamically modulating learning rates based on per-primitive confidence, it achieves a substantial improvement in reconstruction fidelity, boosting the average *PSNR* by +60.2% over the baseline. This confirms that concentrating computational effort on uncertain regions is critical for effective convergence.

Table VI
ABLATION STUDY ON CHALLENGING TUM-RGBD SEQUENCES. WE REPORT RECONSTRUCTION QUALITY (PSNR/SSIM/LPIPS) AND THE PERCENTAGE OF FRAMES PROCESSED RELATIVE TO A DENSE APPROACH.

Method	**Metric**	`2-360`	`2-slam`	Perf. Δ	Frames Proc.
Base	PSNR↑	13.42	17.89	–	100%
	SSIM↑	0.42	0.72	–	
	LPIPS↓	0.54	0.27	–	
+Go-Dial	PSNR↑	25.60	24.55	+60.2%	100%
	SSIM↑	0.90	0.87	+55.3%	
	LPIPS↓	0.20	0.22	-48.1%	
+Go-View	PSNR↑	25.00	24.02	+56.6%	**10.5%**
	SSIM↑	0.87	0.85	+50.9%	
	LPIPS↓	0.25	0.27	-35.8%	
Full (Ours)	PSNR↑	**28.85**	**24.77**	**+71.3%**	**10.5%**
	SSIM↑	**0.94**	**0.89**	**+60.5%**	
	LPIPS↓	**0.11**	**0.20**	**-61.7%**	

The Go-View module is the key to achieving real-time performance. By selectively processing only the most informative 10.5% of input frames, it increases the system's throughput to over 30 *FPS*. While this drastically improves efficiency, the results show a slight drop in quality compared to the +Go-Dial only variant, indicating that reducing the input data volume makes the quality of the subsequent optimization step even more critical.

2) ***Synergistic Effect of the Full System****:* The Full system, integrating all components, demonstrates a clear synergistic effect. It inherits the real-time capability from Go-View while leveraging Go-Dial to maximize reconstruction quality from the sparse but highly informative keyframes. Compared to the Base system, our full approach improves the average *PSNR* by a remarkable +71.3% and reduces perceptual error (*LPIPS*) by -61.7%, all while processing only 10.5% of the frames.

This result supports a crucial insight: strictly filtering for high-quality, informative data (via Ro-NAFNet and Information Gain) is far more effective for 3DGS mapping than blindly integrating all noisy frames. By focusing optimization only on reliable data, the system avoids "polluting" the map with artifacts from blurry or redundant observations. This *"Less is More"* strategy not only ensures real-time performance but also prevents the map from being corrupted, resolving the fundamental trade-off between speed and quality.

G. Evaluation in Real-World Robotic Scenarios

To validate the practical utility and generalization of the maps generated by μSLAM for downstream robotic tasks, we conducted a series of experiments on a real robotic platform.

1) ***Hardware and Scenarios****:* Our experimental platform consists of a Dashgo-D1 wheeled mobile robot equipped with an Intel RealSense D435i RGB-D camera and an onboard computer with an NVIDIA RTX 3090 GPU. We evaluated the system in three representative indoor environments: a furnished living room, a mock unmanned supermarket with narrow aisles and repetitive textures, and a cluttered meeting room. These scenarios were chosen to reflect common

Figure 8. **Real-world robotic evaluation platform and online reconstruction results. (Left Column)** Our experimental hardware consists of a Dashgo-D1 mobile robot equipped with an Intel RealSense D435i sensor for perception and an onboard NVIDIA RTX 3090 GPU for all SLAM computations. **(Right Column)** We show novel view renderings of the final maps built online by μSLAM in three challenging, real-world scenarios: (a) a furnished **living room**, (b) a cluttered **unmanned supermarket**, and (c) a reflective **meeting room**. Despite challenges such as complex geometry, repetitive textures, and variable lighting, our system successfully produces globally consistent, high-fidelity 3D reconstructions suitable for downstream robotic tasks like navigation and scene understanding.

challenges for mobile robots, including complex geometry and potential for perceptual aliasing.

*2) **Experimental Protocol and Results**:* The protocol consists of three stages. First, the robot autonomously navigates a random trajectory to build the map of each environment online using μSLAM. As shown in Fig. 8, high-fidelity scenes were successfully reconstructed under these challenging conditions, achieving an average quality of *PSNR* 21.37 and *SSIM* 0.93.

Second, we demonstrate the map's utility for geometric navigation. The 3D UGF is projected onto the XY plane to generate a 2D occupancy grid. Using this map, we tasked the robot with performing multi-goal navigation using an A* planner [57]. Success was defined as reaching all goals without collision. The high success rate for this task (Tab. VII) serves as a direct validation of the map's geometric accuracy and global consistency.

Third, we evaluate the map's suitability for semantic navigation. We integrate our 3DGS representation with a Vision-Language Model (VLLM) to construct a 3D scene graph in parallel with mapping. This enables the robot to accept natural language commands (e.g., "go to the red sofa"), parse them to identify a target object, and use its 3D location as a goal.

Table VII
SUCCESS RATES FOR DOWNSTREAM ROBOTIC TASKS. WE REPORT SUCCESS RATES FOR GEOMETRIC NAVIGATION, INSTRUCTION PARSING, AND THE FINAL END-TO-END SEMANTIC NAVIGATION TASK ACROSS THREE REAL-WORLD SCENARIOS.

Environment	Geometric Nav.	Instruction Parsing	Semantic Nav.
Living Room	100%	95%	90%
Supermarket	95%	90%	85%
Meeting Room	100%	100%	95%
Average	**98.3%**	**95.0%**	**90.0%**

As reported in Tab. VII, the system achieved a high average end-to-end success rate of 90.0% across all scenarios. These results confirm that the globally consistent, high-fidelity maps produced by μSLAM are not merely visually accurate but also serve as a robust and effective foundation for building advanced spatial intelligence systems on autonomous mobile robots.

V. Conclusion

This paper presents definitive evidence that the prevailing trade-off between real-time performance and reconstruction fidelity in dense SLAM is not fundamental but an artifact of deterministic approaches. Our work, μSLAM, demonstrates a new principle: by explicitly modeling and acting upon per-primitive uncertainty, it is possible to achieve both state-of-the-art accuracy and real-time efficiency simultaneously. This advocates for a paradigm shift in 3DGS-SLAM—from purely geometric reconstruction towards *probabilistically-grounded spatial intelligence*. By endowing the map with a "self-awareness" of its own certainty, we transform it from a static model into an active participant in the state estimation process, enabling systems to reason about ambiguity and robustly navigate the real world. While our implementation represents a principled heuristic, future work should advance this paradigm by exploring formally-derived uncertainty models and adapting these powerful perception capabilities to resource-constrained robotic platforms, thereby democratizing robust, high-fidelity spatial intelligence for a wider range of autonomous systems.

References

[1] D. Maier, A. Hornung, and M. Bennewitz, "Real-time navigation in 3d environments based on depth camera data," in *2012 12th IEEE-RAS International Conference on Humanoid Robots (Humanoids 2012)*, 2012, pp. 692–697.

[2] A. Werby, C. Huang, M. Büchner, A. Valada, and W. Burgard, "Hierarchical open-vocabulary 3d scene graphs for language-grounded robot navigation," in *First Workshop on Vision-Language Models for Navigation and Manipulation at ICRA 2024*, 2024. [Online]. Available: https://openreview.net/forum?id=TL0Hb9OwfR

[3] C. Huang, O. Mees, A. Zeng, and W. Burgard, "Audio visual language maps for robot navigation," in *Experimental Robotics*, M. H. Ang Jr and O. Khatib, Eds. Cham: Springer Nature Switzerland, 2024, pp. 105–117.

[4] Y. Zheng, X. Chen, Y. Zheng, S. Gu, R. Yang, B. Jin, P. Li, C. Zhong, Z. Wang, L. Liu, C. Yang, D. Wang, Z. Chen, X. Long, and M. Wang, "Gaussiangrasper: 3d language gaussian splatting for open-vocabulary robotic grasping," *IEEE Robotics and Automation Letters*, vol. 9, no. 9, pp. 7827–7834, 2024.

[5] S. Zhou, H. Chang, S. Jiang, Z. Fan, Z. Zhu, D. Xu, P. Chari, S. You, Z. Wang, and A. Kadambi, "Feature 3dgs: Supercharging 3d gaussian splatting to enable distilled feature fields," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2024, pp. 21 676–21 685.

[6] R.-Z. Qiu, G. Yang, W. Zeng, and X. Wang, "Feature splatting: Language-driven physics-based scene synthesis and editing," 2024. [Online]. Available: https://arxiv.org/abs/2404.01223

[7] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, "3d gaussian splatting for real-time radiance field rendering," *ACM Transactions on Graphics (ToG)*, vol. 42, no. 4, July 2023.

[8] V. Yugay, Y. Li, T. Gevers, and M. R. Oswald, "Gaussian-slam: Photo-realistic dense slam with gaussian splatting," 2024. [Online]. Available: https://arxiv.org/abs/2312.10070

[9] 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.

[10] H. Matsuki, R. Murai, P. H. J. Kelly, and A. J. Davison, "Gaussian Splatting SLAM," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

[11] L. Zhu, Y. Li, E. Sandström, S. Huang, K. Schindler, and I. Armeni, "Loopsplat: Loop closure by registering 3d gaussian splats," in *2025 International Conference on 3D Vision (3DV)*, 2025, pp. 156–167.

[12] X. Zhong, Y. Pan, L. Jin, M. Popović, J. Behley, and C. Stachniss, "Globally consistent rgb-d slam with 2d gaussian splatting," 2025. [Online]. Available: https://arxiv.org/abs/2506.00970

[13] C. Campos, R. Elvira, J. J. Gomez, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM3: An accurate open-source library for visual, visual-inertial and multi-map SLAM," *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 1874–1890, 2021.

[14] D. Gálvez-López and J. D. Tardós, "Bags of binary words for fast place recognition in image sequences," *IEEE Transactions on Robotics*, vol. 28, no. 5, pp. 1188–1197, October 2012.

[15] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon, "Bundle adjustment — a modern synthesis," in *Vision Algorithms: Theory and Practice*, B. Triggs, A. Zisserman, and R. Szeliski, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2000, pp. 298–372.

[16] B. Curless and M. Levoy, "A Volumetric Method for Building Complex Models from Range Images," 1996. [Online]. Available: http://papers.cumincad.org/data/works/att/2ca3.content.pdf

[17] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohli, J. Shotton, S. Hodges, and A. Fitzgibbon, "KinectFusion: Real-Time Dense Surface Mapping and Tracking," 2011. [Online]. Available: https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/ismar2011.pdf

[18] T. Whelan, M. Kaess, M. Fallon, H. Johannsson, J. Leonard, and J. McDonald, "Kintinuous: Spatially Extended KinectFusion," in *Proc. of the RSS Workshop on RGB-D: Advanced Reasoning with Depth Cameras*, 2012. [Online]. Available: http://www.thomaswhelan.ie/Whelan12rssw.pdf

[19] A. Dai, M. Nießner, M. Zollhöfer, S. Izadi, and C. Theobalt, "BundleFusion: Real-time Globally Consistent 3D Reconstruction using Online Surface Re-integration," vol. 36, no. 3, pp. 1–18, 2017. [Online]. Available: https://arxiv.org/pdf/1604.01093v1.pdf

[20] E. Palazzolo, J. Behley, P. Lottes, P. Giguere, and C. Stachniss, "ReFusion: 3D Reconstruction in Dynamic Environments for RGB-D Cameras Exploiting Residuals," 2019. [Online]. Available: proceedings:palazzolo2019iros.pdf

[21] I. Vizzo, T. Guadagnino, J. Behley, and C. Stachniss, "VDBFusion: Flexible and Efficient TSDF Integration of Range Sensor Data," vol. 22, no. 3, p. 1296, 2022. [Online]. Available: https://www.mdpi.com/1424-8220/22/3/1296/pdf

[22] F. Endres, J. Hess, J. Sturm, D. Cremers, and W. Burgard, "3D Mapping with an RGB-D Camera," vol. 30, no. 1, pp. 177–187, 2014. [Online]. Available: http://ais.informatik.uni-freiburg.de/publications/papers/endres14tro.pdf

[23] A. Hornung, K. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "OctoMap: An Efficient Probabilistic 3D Mapping Framework Based on Octrees," vol. 34, no. 3, pp. 189–206, 2013. [Online]. Available: http://www.informatik.uni-freiburg.de/~stachnis/pdf/hornung13auro.pdf

[24] J. Stückler and S. Behnke, "Multi-Resolution Surfel Maps for Efficient Dense 3D Modeling and Tracking," vol. 25, no. 1, pp. 137–147, 2014. [Online]. Available: https://www.ais.uni-bonn.de/papers/JVCI_13_RGB-D-SLAM.pdf

[25] T. Whelan, S. Leutenegger, R. S. Moreno, B. Glocker, and A. Davison, "ElasticFusion: Dense SLAM Without A Pose Graph," 2015. [Online]. Available: proceedings:whelan2015rss.pdf

[26] M. Keller, D. Lefloch, M. Lambers, and S. Izadi, "Real-time 3D Reconstruction in Dynamic Scenes using Point-based Fusion," 2013. [Online]. Available: http://reality.cs.ucl.ac.uk/projects/kinect/keller13realtime.pdf

[27] 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," 2022. [Online]. Available: https://arxiv.org/pdf/2112.12130.pdf

[28] X. Yang, H. Li, H. Zhai, Y. Ming, Y. Liu, and G. Zhang, "Vox-Fusion: Dense Tracking and Mapping with Voxel-based Neural Implicit Representation," 2022. [Online]. Available: https://arxiv.org/pdf/2210.15858.pdf

[29] H. Wang, J. Wang, and L. Agapito, "Co-SLAM: Joint Coordinate and Sparse Parametric Encodings for Neural Real-Time SLAM," 2023. [Online]. Available: https://arxiv.org/pdf/2304.14377

[30] M. M. Johari, C. Carta, and F. Fleuret, "ESLAM: Efficient Dense SLAM System Based on Hybrid Representation of Signed Distance Fields," 2023. [Online]. Available: https://publications.idiap.ch/attachments/papers/2023/Johari_CVPR_2023.pdf

[31] T. Deng, G. Shen, T. Qin, J. Wang, W. Zhao, J. Wang, D. Wang, and W. Chen, "PLGSLAM: Progressive Neural Scene Represenation with Local to Global Bundle Adjustment," 2024. [Online]. Available: https://openaccess.thecvf.com/content/CVPR2024/papers/Deng_PLGSLAM_Progressive_Neural_Scene_Represenation_with_Local_to_Global_Bundle_CVPR_2024_paper.pdf

[32] E. Sandström, Y. Li, L. Van Gool, and M. R. Oswald, "Point-SLAM: Dense Neural Point Cloud-based SLAM," 2023. [Online]. Available: https://arxiv.org/pdf/2304.04278.pdf

[33] L. Liso, E. Sandström, V. Yugay, L. Van Gool, and M. R. Oswald, "Loopy-slam: Dense neural slam with loop closures," 2024. [Online]. Available: https://arxiv.org/pdf/2402.09944

[34] G. Zhang, E. Sandström, Y. Zhang, M. Patel, L. Van Gool, and M. R. Oswald, "GlORIE-SLAM: Globally Optimized Rgb-only Implicit Encoding Point Cloud SLAM," vol. arXiv:2403.19549, 2024. [Online]. Available: https://arxiv.org/pdf/2403.19549

[35] H. Matsuki, R. Murai, P. H. Kelly, and A. J. Davison, "Gaussian splatting SLAM," 2024. [Online]. Available: https://arxiv.org/pdf/2312.06741

[36] C. Yan, D. Qu, D. Xu, B. Zhao, Z. Wang, D. Wang, and X. Li, "GS-SLAM: Dense Visual SLAM with 3D Gaussian Splatting," 2024. [Online]. Available: https://arxiv.org/pdf/2311.11700

[37] 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," 2024. [Online]. Available: https://arxiv.org/pdf/2312.02126

[38] V. Yugay, Y. Li, T. Gevers, and M. R. Oswald, "Gaussian-SLAM: Photo-realistic Dense SLAM with Gaussian Splatting," vol. arXiv:2312.10070, 2023. [Online]. Available: http://arxiv.org/pdf/2312.10070

[39] S. Sun, M. Mielle, A. J. Lilienthal, and M. Magnusson, "High-Fidelity SLAM Using Gaussian Splatting with Rendering-Guided Densification and Regularized Optimization," 2024. [Online]. Available: https://arxiv.org/pdf/2403.12535

[40] E. Giacomini, L. Di Giammarino, L. D. Rebott, G. Grisetti, and M. R. Oswald, "Splat-LOAM: Gaussian Splatting LiDAR Odometry and Mapping," vol. arXiv:2503.17491, 2025. [Online]. Available: https://arxiv.org/pdf/2503.17491

[41] S. Thrun, "Probabilistic robotics," *Communications of the ACM*, vol. 45, no. 3, pp. 52–57, 2002.

[42] C.-M. Chung, Y.-C. Tseng, Y.-C. Hsu, X.-Q. Shi, Y.-H. Hua, J.-F. Yeh, W.-C. Chen, Y.-T. Chen, and W. H. Hsu, "Orbeez-slam: A real-time monocular visual slam with orb features and nerf-realized mapping," in *ICRA*, 2023, pp. 9400–9406.

[43] S. Ha, J. Yeon, and H. Yu, "Rgbd gs-icp slam," in *ECCV*. Springer, 2025, pp. 180–197.

[44] H. Huang, L. Li, C. Hui, and S.-K. Yeung, "Photo-slam: Real-time simultaneous localization and photorealistic mapping for monocular, stereo, and rgb-d cameras," in *CVPR*, 2024.

[45] L. Zhu, Y. Li, E. Sandström, S. Huang, K. Schindler, and I. Armeni, "LoopSplat: Loop Closure by Registering 3D Gaussian Splats," 2025. [Online]. Available: https://arxiv.org/pdf/2408.10154

[46] V. Leroy, Y. Cabon, and J. Revaud, "Grounding image matching in 3d with mast3r," 2024.

[47] L. Chen, X. Chu, X. Zhang, and J. Sun, "Simple baselines for image restoration," *arXiv preprint arXiv:2204.04676*, 2022.

[48] D. Barath, J. Noskova, M. Ivashechkin, and J. Matas, "Magsac++, a fast, reliable and accurate robust estimator," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[49] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of rgb-d slam systems," in *Proc. of the International Conference on Intelligent Robot Systems (IROS)*, Oct. 2012.

[50] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner, "Scannet: Richly-annotated 3d reconstructions of indoor scenes," in *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017.

[51] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "G2o: A general framework for graph optimization," in *2011 IEEE International Conference on Robotics and Automation*, 2011, pp. 3607–3613.

[52] R. Mur-Artal and J. D. Tardós, "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras," *IEEE transactions on robotics (TRO)*, vol. 33, no. 5, pp. 1255–1262, 2017.

[53] M. Labbe and F. Michaud, "RTAB-Map: An open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation," vol. 36, no. 1, pp. 416–446, 2019. [Online]. Available: https://onlinelibrary.wiley.com/doi/pdf/10.1002/rob.21831

[54] Y. Zhang, F. Tosi, S. Mattoccia, and M. Poggi, "Go-slam: Global optimization for consistent 3d instant reconstruction," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2023.

[55] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," vol. 13, no. 4, pp. 600–612, 2004. [Online]. Available: https://www.cns.nyu.edu/pub/lcv/wang03-preprint.pdf

[56] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," 2018.

[57] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968.