

Article

Lightweight 3DGS-SLAM for Memory-Constrained Environments: Spatial-Aware Truncation and Adaptive Antihallucination Restoration Mechanism

Honghui Fan ^{1,†}, Zikai Li ^{1,†}, Hongjin Zhu ¹ and Wenhe Chen ^{2,*}

¹ School of Computer Engineering, Jiangsu University of Technology, Changzhou 213001, China; fanhonghui@jsut.edu.cn (H.F.); 2024655204@smail.jsut.edu.cn (Z.L.); zhuhongjin@jsut.edu.cn (H.Z.)

² School of Internet of Things Engineering, Jiangnan University, Wuxi 214122, China

* Correspondence: chenwh@jsut.edu.cn

† These authors contributed equally to this work.

Highlights

What are the main findings?

- The paper proposes SATA-SLAM, a lightweight 3DGS-SLAM framework for memory-constrained environments. It uses a Spatial-Aware Pruning Module (SAPM) for the front-end and an Anti-Hallucination Generative Refinement Module (AGRM) for the back-end.
- Under a strict memory constraint (0.8 M primitives, ~26% of original memory), it achieves high-fidelity rendering (PSNR 37.44 dB), markedly outperforming naive pruning (12.5 dB) and maintaining tracking accuracy.

What are the implications of the main findings?

- It solves the severe memory bottleneck in long-sequence 3DGS-SLAM, enabling deployment on resource-constrained platforms
- It proves the effectiveness of combining physical geometric constraints with large-model generative priors, successfully solving the trade-off between resource efficiency and perceptual fidelity while mitigating AIGC hallucination risks.

Abstract

Dense simultaneous localization and mapping (SLAM) via 3D Gaussian splatting (3DGS) faces memory bottlenecks due to the explosive growth of primitives during long-sequence mapping. We propose SATA-SLAM, a framework featuring spatial-aware truncation and adaptive anti-hallucination. The online front-end maintains a constant memory footprint via a spatial-aware pruning module (SAPM), which employs a survival scoring function that couples primitive opacity with view-frustum projection coverage and a temporal protection window. Subsequently, an anti-hallucination generative refinement module (AGRM) utilizes texture priors from pretrained diffusion models for offline inpainting of residual regions. In addition, an adaptive gating mechanism to verify and suppress AIGC-induced hallucinations caused by pose drift, ensuring multiview consistency. Experiments on the public Replica dataset show that SATA-SLAM improves rendering quality from 12.5 dB to 37.44 dB (averaged over the Replica room0 and office0 scenes) while using only 26% of the original memory, outperforming the unconstrained baseline. This study provides a pathway toward low-power, high-fidelity environmental perception for mobile robots.

Academic Editors: Wolfgang Kainz, Junxing Zheng and Peng Cao

Received: 21 May 2026

Revised: 3 July 2026

Accepted: 6 July 2026

Published: 6 July 2026

Copyright: © 2026 by the authors.

Published by MDPI on behalf of the International Society for Photogrammetry and Remote Sensing. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\) license](https://creativecommons.org/licenses/by/4.0/).

Keywords: 3D Gaussian splatting; SLAM; memory optimization; spatial-aware pruning; antihallucination restoration

1. Introduction

Simultaneous localization and mapping (SLAM) provides fundamental support for robots to achieve autonomous navigation, environmental perception, and digital twin tasks. In recent years, the field of dense RGB-D SLAM has experienced a technical paradigm shift. From early TSDF explicit representations to neural implicit representations represented by iMAP and NICE-SLAM [1,2], researchers have continuously pursued higher-precision scene reconstruction. However, implicit representations are often limited by high coordinate sampling overhead, making it difficult to achieve real-time rendering.

Recently, the emergence of three-dimensional Gaussian splatting (3DGS) [3] has led to breakthroughs in this field. By decoupling the scene into millions of explicit Gaussian primitives, 3DGS achieves extremely high inference speeds while ensuring photorealistic rendering quality. In the context of remote sensing, 3DGS has been increasingly deployed for UAV-based mapping [4] and multi-temporal satellite image synthesis [5] due to its superior efficiency compared to traditional implicit models. Advanced methods represented by SplaTAM [6], through online optimization of 3D Gaussian maps, have provided unprecedented fidelity in indoor scene reconstruction. However, such high fidelity often comes at the cost of resource explosion [7]. In the continuous mapping process, the number of Gaussian primitives increases linearly or even exponentially, typically exceeding 3 million within a few minutes. For mobile robots or embedded platforms with limited memory capacity, this unconstrained memory consumption has become a core bottleneck restricting their large-scale application [8].

To address these challenges, the most straightforward strategy is to implement fixed-capacity primitive truncation. However, our preliminary experiments revealed that under memory-constrained conditions, simple naive pruning, due to its lack of global topology perception, results in the indiscriminate deletion of a large number of key primitives supporting far-view structures. This causes severe geometric fragmentation and structural collapse. The experimental results demonstrate that the use of this scheme leads to a sharp decrease in PSNR to 12.5 dB and a notable increase in ATE RMSE to 12.40 cm, causing the SLAM method to lose all usability, as shown in Figure 1. This deep contradiction between accuracy and capacity leads to the following two core scientific questions. First, how can the most topology-critical primitives be retained under strict resource constraints through the use of adaptive selection strategies? Second, under the premise of a limited number of physical primitives, can generative priors be employed to compensate for missing details and effectively prevent the inherent hallucination risk of AIGC?

To address the aforementioned challenges, a spatial-aware truncation and adaptive antihallucination SLAM (SATA-SLAM) framework is proposed in this paper. Unlike existing methods, the core of this research lies in ensuring a constant memory upper limit for the front-end SLAM system, which enables long-term deployment on mobile platforms and controlled generative restoration at the back end. Specifically, we introduce a primitive survival evaluation strategy coupled with view-frustum geometry at the front end, complemented by a temporal protection window, to maintain the structural baseline under capacity compression. At the back end, we leverage diffusion priors for conditional inpainting of residual regions and realize the selective adoption of generation results through a perception-driven validation mechanism. The subsequent sections provide a review of the research progress in related fields, an elaboration of the design details and

mathematical principles of the proposed modules, and a verification of the effectiveness of this framework through systematic experiments based on standard datasets.



Figure 1. Naive pruning.

2. Related Work

Dense 3DGS-SLAM technology, as an explicit volumetric representation method, is rapidly reshaping the research landscape of dense visual SLAM because of its superior rendering quality and real-time differentiable optimization capabilities. As foundational research in this field, SplatAM [6] and Splat-SLAM [9] first demonstrated the immense potential of integrating 3DGS into dense RGB-D SLAM systems, achieving photorealistic rendering speeds up to 400 FPS through online optimization of Gaussian primitives. Compared with neural implicit representation methods that rely on a multilayer perceptron (MLP) (such as NICE-SLAM [2]), 3DGS can effectively prevent catastrophic forgetting [10,11] and provide more accurate explicit spatial boundary perception. Subsequent methods such as GS-SLAM [12] and MonoGS [13] extended this paradigm to more diverse application scenarios. However, most existing methods rely on unconstrained growth strategies, leading to an exponential increase in the number of Gaussian primitives during indoor long-sequence scanning (typically millions), which not only imposes a high memory burden on mobile platforms but also limits the real-time performance and scalability of the system because of an excessive number of redundant computations.

To overcome memory capacity bottlenecks, researchers have proposed various compression and pruning strategies for 3DGS, aiming to increase the rendering efficiency by reducing model complexity. Some methods aim to filter redundant primitives by monitoring opacity or 3D scaling parameters [14]; in addition, methods such as LightGaussian [15,16] exclude components with low contributions to the final rendering by introducing saliency detection mechanisms. However, existing strategies focus mostly on offline compression of static scenes. In real-time incremental mapping environments such as SLAM, simple global threshold pruning often lacks dynamic perception of the view-frustum spatial layout. Under extreme compression ratios [17,18], these methods that lack local geometric sensitivity can easily cause fragmentation of scene topology and the occurrence of rendering holes.

Recently, the field of 3DGS-SLAM has witnessed markedly advances across multiple directions. In omnidirectional perception, ODGS-SLAM [19] extends 3DGS-SLAM to panoramic image inputs. In dynamic environment handling, DGS-SLAM [20] introduces Gaussian splatting into dynamic SLAM for the first time, achieving robust localization in non-static environments through dynamic object handling mechanisms. In uncertainty modeling, CG-SLAM [21] proposes a dense RGB-D SLAM system based on uncertainty-

aware 3D Gaussian fields. In memory optimization, QCG-SLAM [22] markedly reduces memory footprint through a compact Gaussian representation enabled by quadtree partitioning. Unlike these works, which focus respectively on omnidirectional extension, dynamic processing, uncertainty modeling, or memory compression, this work concentrates on achieving a constant memory upper bound for front-end SLAM via spatially aware pruning, providing a new solution for deployment on memory-constrained mobile platforms while maintaining localization accuracy. In dynamic environment handling, DGS-SLAM [20] first introduces Gaussian splatting into dynamic SLAM, handling dynamic objects through robust filtering and cross-keyframe photometric consistency constraints. DynaGS-SLAM [23] identifies and excludes dynamic regions through semantic segmentation. Unlike these methods, which focus on dynamic scene processing, this work targets memory efficiency optimization in static indoor scenes. Robustness extension to dynamic scenes is left as future work.

The use of large-scale pretrained generative models (such as diffusion models) [24,25] as priors provides a new pathway for addressing hole filling and texture restoration in three-dimensional reconstruction. Researchers have attempted to introduce generative restoration into the back ends of NeRF or 3DGS [26–28] to compensate for blind spots in sensor scanning. Specifically, generative priors can provide fine detail enhancement for generating sparse maps under memory constraints through their notable distribution modeling capabilities. However, in the dynamic operation pipeline of SLAM [29], camera pose drift and sensor noise markedly amplify the uncertainty in generative models, thereby inducing severe hallucination risks and producing multiview inconsistency artifacts. Currently, how to robustly integrate generative priors into real-time SLAM systems and effectively intercept hallucinated outputs that reduce global map consistency remains a challenging open subject. The adaptive gating mechanism proposed in this paper intervenes through real-time perception of epistemic uncertainty, thus fundamentally ensuring generated logical coherence and physical authenticity.

3. Materials and Methods

In this paper, a decoupled end-cloud lightweight 3DGS-SLAM framework, namely, SATA-SLAM, is proposed to achieve high-fidelity indoor scene reconstruction under strict memory-constrained conditions. SATA-SLAM comprises two main collaborative parts. The first part is the online front end running on the local end, which constrains the total number of Gaussian primitives within a preset upper bound through a spatial-aware truncation strategy to ensure that the system exhibits a constant and extremely low memory footprint. The second part is the offline back end deployed on high-performance computing nodes, which employs texture priors from pretrained diffusion models for generative inpainting of residual regions caused by truncation, supplemented by an adaptive gating mechanism for anti-hallucination validation. The core advantage of this framework lies in the effective decoupling of local-end and cloud-end computing power. This approach not only overcomes the problems of geometric fragmentation and structural collapse caused by traditional naive pruning and eliminates multiview artifacts and hallucination pollution in generative restoration but also achieves high-fidelity map reconstruction under extremely severe resource constraints (occupying only approximately 26.7% of the memory parameters of the original systems). A flow chart is shown in Figure 2. The following subsections provide explanations of the design principles and mathematical formulations of the aforementioned core modules.

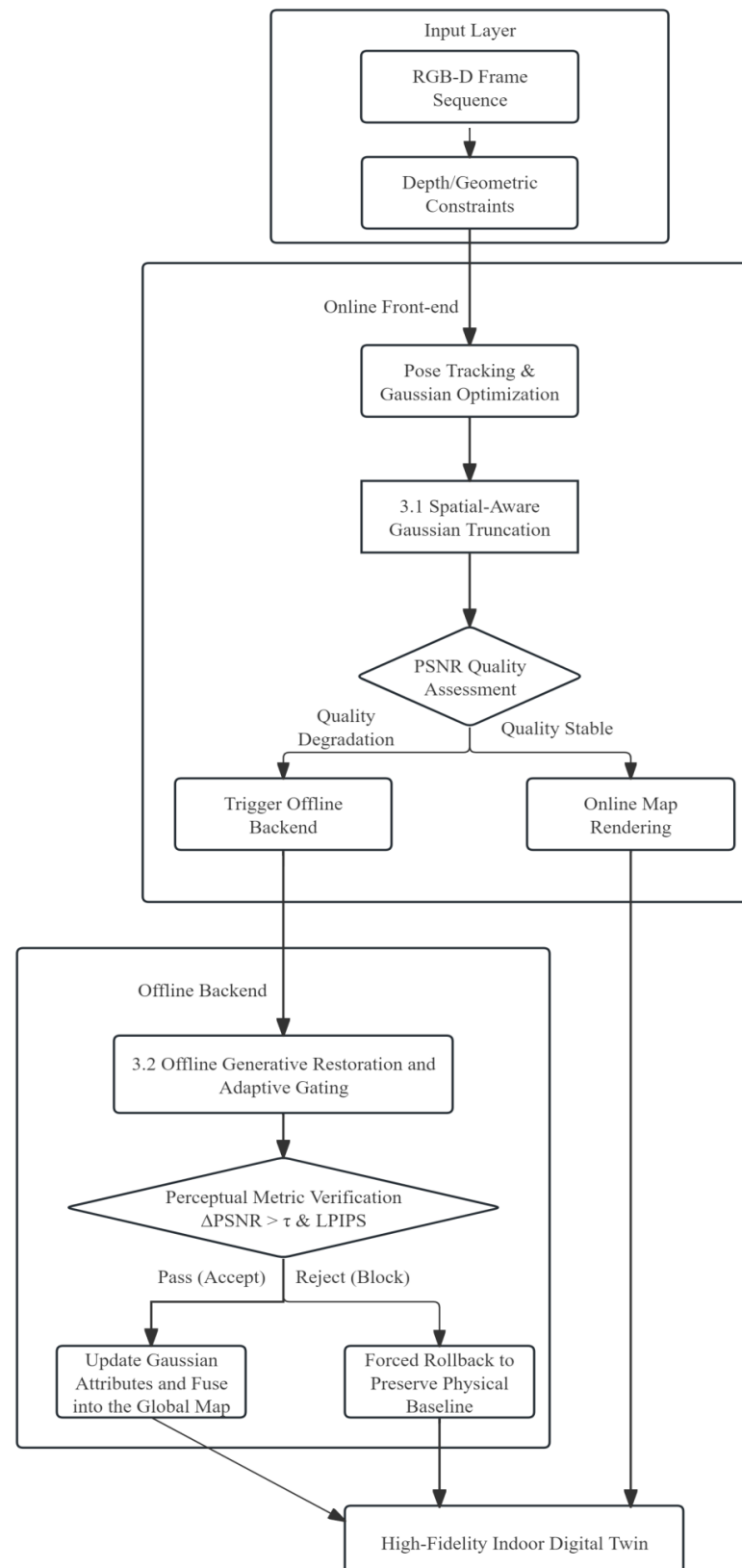


Figure 2. Overall flow chart.

3.1. Online Front-End: Spatial-Aware Gaussian Truncation

The online front-end inherits the basic tracking and mapping pipeline of SplatAM, which is responsible for real-time camera pose estimation and online optimization of Gaussian primitives. On this basis, we introduce a spatial-aware truncation module,

which constrains the total number of Gaussian primitives within a preset upper bound $N_{\max}$ (0.8 M) through the use of a survival scoring mechanism coupled with view-frustum geometric saliency, thereby ensuring a constant memory footprint for the system on resource-constrained platforms, as shown in Figure 3

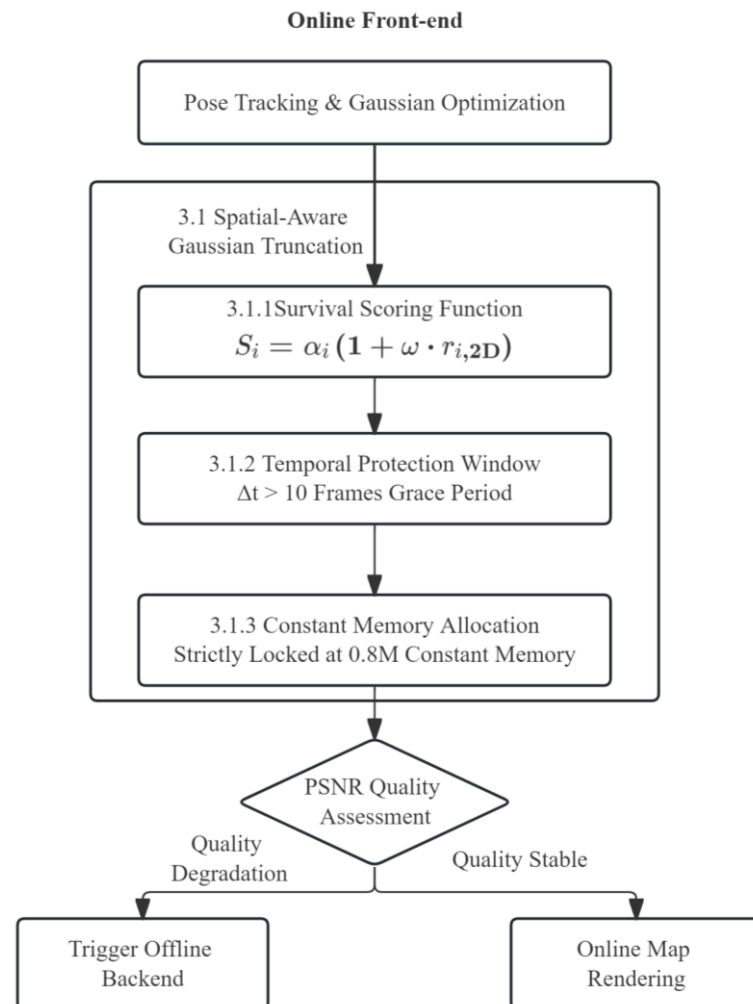


Figure 3. Online front-end flow chart.

3.1.1. Survival Scoring Function

After the optimization iteration of each frame is completed, the system aims to evaluate the structural importance of all the active Gaussian primitives. Specifically, the survival score S_i of the i -th primitive is defined as follows:

$$S_i = o_i \cdot (1 + \omega \cdot \bar{r}_{i,2D}) \quad (1)$$

where $o_i \in [0,1]$ is the opacity of the primitive after sigmoid activation, and $\bar{r}_{i,2D}$ is the normalized radius of the primitive after being projected onto the image plane within the current view-frustum [30]. The hyperparameter ω is used to adjust the contribution weight of the geometric scale to the survival score. An intuitive explanation of this equation is that although large-area background structures (such as walls and floors) may exhibit low opacity gradients, their 2D projection coverage is high. Through compensation of the $\bar{r}_{i,2D}$ term, the system can prioritize the retention of these key primitives that support the global topology of the scene, thereby effectively avoiding the occurrence of structural holes caused by traditional threshold truncation.

The choice of a linear weighted formulation is motivated by three considerations. Interpretability—opacity and projected radius respectively represent the per-view contribution strength and the spatial coverage of the scene, both possessing clear physical meaning. Computational efficiency—linear operations incur minimal overhead in real-time SLAM. Empirical validity—ablation studies (Section 4.3) demonstrate that $\omega = 1.0$ yields optimal performance, and more complex functional forms do not necessarily provide additional benefits. The physical intuition behind this scoring function capturing “topological importance” is as follows: a Gaussian covering a large area (e.g., a wall) with low opacity provides far greater structural support than one covering only a few pixels with high opacity. Multiplying the two factors accounts for both visual contribution and spatial coverage, jointly measuring the overall supportive capacity of a primitive to the scene topology.

3.1.2. Temporal Protection Window

In the continuous exploration process of SLAM, newly generated Gaussian primitives often require several frames of gradient updates to converge to a stable geometric form. If these primitives are eliminated by the scoring mechanism before convergence is reached, the geometric reconstruction quality of newly explored areas will deteriorate markedly. Therefore, we introduce a simple and effective temporal protection mechanism. Notably, for primitives with a generation time smaller than $\Delta t = 10$ frames, the system directly grants them survival immunity, and they do not participate in the current round of top-k filtering. We validate the rationale behind this choice through ablation experiments (detailed in Section 4.3). The results demonstrate that $\Delta t = 10$ achieves the optimal balance between protecting the convergence of newly added Gaussians and promptly releasing redundant capacity.

3.1.3. Constant Memory Allocation

Combining survival scoring and temporal protection, the system filters out and retains the $N_{\max}$ primitives with the highest scores through the top-k operator in each frame:

$$\mathcal{G}_t = \{g_i \mid S_i \in \text{Top} - K(S, N_{\max})\} \quad (2)$$

where $\mathcal{G}_t$ denotes the set of active 3D Gaussian primitives ultimately retained by the system after the t -th frame is processed; g_i denotes the i -th individual Gaussian primitive in the scene; and S_i denotes the survival score of the i -th primitive calculated based on the spatial-aware scoring function. Moreover, S denotes the total set of survival scores of all active primitives participating in filtering (i.e., not within the temporal protection period) in the current frame; $N_{\max}$ is the strict upper bound (capacity threshold) of the total number of Gaussian primitives preset by the system based on hardware constraints; and $\text{Top} - K(S, N_{\max})$ denotes the operation operator that extracts the top $N_{\max}$ highest scores from the score set S .

This mechanism mathematically guarantees a strict upper bound for the number of active primitives, thereby endowing the system with a constant memory upper bound. In the experiments, we set $N_{\max}$ to 0.8 M, which is approximately 26% of the peak capacity of the original SplaTAM in the same sequence. The measured peak GPU memory consumption of the front end is 1.22 GB (TUM), converging to a stable value once the mapping window is reached and exhibiting no unbounded growth with the number of frames.

3.1.4. Front-End Optimization Objective

In the process of online tracking and mapping, the system iteratively optimizes the spatial position, rotation, scale, and opacity attributes of Gaussian primitives so that the rendered image continuously approaches the input ground-truth frame in terms of color,

texture, and geometric features. This process aims to establish precise geometric constraints to maintain subcentimeter-level pose tracking and provide the initial construction of the topological structure of the scene. For detailed mathematical definitions of the photometric losses and geometric constraints involved at the front-end stage, please refer to Section 3.3.1 of this chapter.

3.2. Offline Back-End: Offline Generative Restoration and Adaptive Gating

The offline back end is activated when the rendering quality decreases at the front end because of capacity constraints. This back end relies on texture priors from pretrained diffusion models to conduct generative inpainting for residual regions. It is important to clarify that AGRM operates as an offline post-processing module, initiated only after the complete SLAM trajectory estimation has been finalized. Its refinement results are not fed back to the front-end camera tracking or mapping modules in any form. Consequently, regardless of whether the gating mechanism is enabled, AGRM does not affect camera trajectory estimation (ATE). This design ensures the stability of the front-end real-time SLAM and the feasibility of lightweight deployment. The GPU memory consumption of its diffusion model and feature cache is not counted toward the front-end real-time memory budget and can be executed asynchronously on high-performance. To guard against the inherent hallucination risk of generative models, we design an adaptive gating mechanism to implement real-time validation of the restoration results through PSNR-based perceptual validation, merging them into a global map only when a positive gain is confirmed. After truncation is completed by the online front end, areas covered by partially eliminated primitives exhibit texture loss and decreased rendering quality. To compensate for this loss, the method activates the offline back end at keyframes for refined inpainting, as shown in Figure 4.

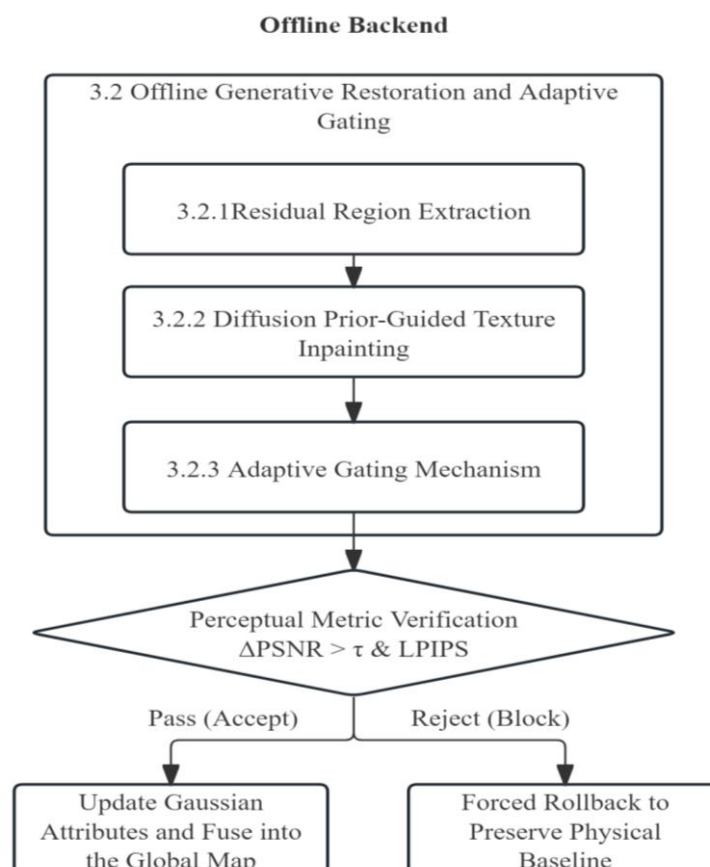


Figure 4. Offline back-end flow chart.

3.2.1. Residual Region Extraction

We extract a precise binary inpainting mask $M \in \{0,1\}^{H \times W}$ to localize geometric and textural holes caused by primitive truncation. The extraction pipeline comprises the following four steps:

RGB residual computation. We compute the per-pixel absolute difference between the rendered image and the ground-truth image, taking the maximum across the RGB channels as the residual response:

$$D_{rgb}(u, v) = \max_{c \in R, G, B} |I_{render}^c(u, v) - I_{GT}^c(u, v)| \quad (3)$$

where (u, v) denotes pixel coordinates and c is the color channel index.

Adaptive thresholding. We threshold the residual map to obtain an initial binary mask:

$$M_{init}(u, v) = 1(D_{rgb}(u, v) > \tau_{rgb}) \quad (4)$$

where $1(\cdot)$ is the indicator function. The threshold τ_{rgb} adopts an adaptive strategy: the default value is 30 (pixel value range 0–255), reduced to 20 for sparsely textured regions (e.g., white walls) and increased to 50 for densely textured regions (e.g., cluttered desk-tops). In practice, $\tau_{rgb} \in [20, 50]$ proves robust across all tested scenes.

Edge-aware morphological processing. To prevent the mask from crossing strong geometric boundaries—which would cause the inpainting model to blend content from different objects—we apply Canny edge detection on the ground-truth image:

$$M_{edge}(u, v) = \text{Canny}(I_{GT}(u, v), \tau_{low} = 50, \tau_{high} = 150) \quad (5)$$

Subsequently, we perform morphological closing with a 15×15 kernel to fill small holes and smooth boundaries, followed by constraining with the edge mask:

$$M_{closed} = \text{Close}(M_{init}, K_{15 \times 15}) \quad (6)$$

$$M = M_{closed} \cap \overline{M_{edge}}$$

where $\text{Close}(\cdot, K)$ denotes morphological closing with structuring element K , $\overline{M_{edge}}$ denotes the complement of the edge mask (i.e., non-edge regions), and $\cap$ denotes per-pixel logical AND.

Depth consistency verification. Since our data loading pipeline already filters out frames without valid depth maps via `associate.txt` alignment (Section 3.4), the depth-conditioned ControlNet receives input only on frames where depth maps exist. Consequently, mask extraction operates exclusively on geometrically valid regions. Frames with invalid depth are skipped during the inpainting pipeline. The complete extraction procedure is summarized in Algorithm 1.

Algorithm 1. Extraction procedure

Step	Operation/Formula	Description
Input	I_r, I_{gt}	Render image, Ground truth
Output	M	Binary mask
1	$\tau_{rgb} \leftarrow 30, k \leftarrow 15, \tau_{low} \leftarrow 50, \tau_{high} \leftarrow 150$	Parameter initialization
2	$D_{rgb}(u, v) \leftarrow \max_c I_r^c(u, v) - I_{gt}^c(u, v) $	Compute RGB residual
3	$M_{init}(u, v) \leftarrow 1(D_{rgb}(u, v) > \tau_{rgb})$	Adaptive thresholding
4	$M_{closed} \leftarrow \text{Close}(M_{init}, K_{k \times k})$	Closing to fill holes
5	$M_{edge} \leftarrow \text{Canny}(I_{gt}, \tau_{low}, \tau_{high})$	Edge detection
6	$M \leftarrow M_{closed} \cap \overline{M_{edge}}$	Edge-immune masking
7	return M	Return binary mask

3.2.2. Diffusion Prior-Guided Texture Inpainting

When the system detects a decrease in rendering quality caused by primitive truncation, the restoration module based on diffusion priors is activated. This module sends the residual rendered image and the corresponding mask to a pretrained conditional diffusion model [31], thereby employing texture priors learned from large-scale image data to conduct semantic-level generative inpainting for the residual regions. Similarly, ICDDPM proposed by Zhao et al. [32] employs 2D images as conditional information to guide diffusion models, achieving single-view reconstruction of complex point clouds from sparse observations. This provides cross-domain theoretical support for our design of using depth maps as geometric constraints to guide diffusion-based inpainting.

In the restoration process, the model uses the rendered depth of the current frame as a geometric constraint to ensure structural consistency between the generated content and the original scene. For details on the optimization objective function and loss weight settings of the diffusion model in the inpainting process, please refer to Section 3.3.2. Discussion on depth noise propagation. The rendered depth maps produced by the front-end sparse truncation inevitably contain holes and noise due to the primitive capacity limit (0.8 M). These defects may propagate to the inpainting results via the depth-conditioned ControlNet. Nevertheless, the current system remains capable of producing usable results for the following reasons: the pretrained encoder of ControlNet exhibits a certain degree of tolerance to input noise; inpainting is restricted to local regions indicated by the residual mask, thereby confining the scope of noise influence; and the gating mechanism effectively filters out low-quality inpainting caused by depth noise. To address this issue more systematically, potential mitigation strategies include depth map preprocessing (median filtering, morphological closing), confidence-weighted conditioning, and multi-view depth fusion. We leave the systematic exploration of these strategies as future work.

3.2.3. Adaptive Gating Mechanism

Considering the inherent hallucination risk [33] of generative models, the system introduces an adaptive gating mechanism to intercept multiview inconsistent outputs. This

mechanism determines whether to adopt the current generation result by evaluating the rendering gain of the image before and after restoration in real time [34].

The gating module only merges the restored texture features into the global Gaussian map once real image quality improvement is identified and physical consistency is not violated. This interception mechanism fundamentally eliminates the pollution of SLAM system trajectory tracking and map consistency by false generated content.

3.3. System Multistage Optimization Objectives and Loss Functions

The SATA-SLAM framework encompasses two independent operating stages, i.e., online real-time mapping and offline generative restoration. The overall optimization objective of the system is not a single static loss but comprises front-end geometric–photometric joint constraints and back-end generative prior constraints at different stages.

3.3.1. Front-End Online Tracking and Mapping Loss

At the online operation stage of the system, to ensure that Gaussian primitives can accurately fit the physical structure of the scene, the front-end system optimizes the 3DGS parameters in real time by minimizing the residual between the rendered images and the input ground truth. The front-end optimization objective $\mathcal{L}_{\text{frontend}}$ comprises color loss and depth loss weighted together, which can be expressed as:

$$\mathcal{L}_{\text{frontend}} = \lambda_{\text{rgb}} \mathcal{L}_{\text{rgb}} + \lambda_{\text{depth}} \mathcal{L}_{\text{depth}} \quad (7)$$

where λ_{rgb} and λ_{depth} are the hyperparameter weights for color loss and depth loss, respectively. The depth loss $\mathcal{L}_{\text{depth}}$ ensures the geometric accuracy of the physical primitives, while the color loss $\mathcal{L}_{\text{rgb}}$ combines the L_1 norm and the structural similarity (SSIM) loss:

$$\mathcal{L}_{\text{rgb}} = (1 - \alpha) \mathcal{L}_1 + \alpha \mathcal{L}_{\text{SSIM}} \quad (8)$$

where α is a balancing coefficient. The balancing coefficient α in Equation (8) is empirically tuned to trade off pixel-level accuracy (L_1) and structural similarity (SSIM), with the final value selected based on rendering quality across representative scenes. Through the aforementioned explicit front-end objective function, the system achieves real-time sub-centimeter-level pose tracking and basic scene reconstruction.

3.3.2. Back-End Generative Restoration Loss

When the rendering quality of the system decreases because of capacity constraints (e.g., the 0.8 M truncation limit is reached), the offline back end is activated. At this point, front-end physical optimization is paused, and the system instead employs pretrained diffusion model texture priors for texture inpainting of the residual regions. To ensure that the generated results possess realistic textures while maintaining geometric consistency, the core optimization objective of this stage follows the conditional denoising loss function of latent diffusion models (LDMs):

$$\mathcal{L}_{\text{LDM}} = \mathbb{E}_{\epsilon, z_t, t, C} [\| \epsilon - \epsilon_{\theta}(z_t, t, C) \|_2^2] \quad (9)$$

where z_t is the noisy image in the latent space, t is the diffusion time step, ϵ denotes the real noise, and ϵ_{θ} is the noise predicted by the network. Moreover, C is the conditional feature (e.g., depth map guidance feature). In summary, the front-end loss $\mathcal{L}_{\text{frontend}}$ is responsible for establishing the physical consistency baseline, while the back-end loss $\mathcal{L}_{\text{LDM}}$ is responsible for ensuring visual enhancement beyond the upper bound of the physical representation under elemental constraints. The two act alternately on the time axis, jointly constituting the complete optimization system of the SATA-SLAM framework.

3.4. Experimental Settings

3.4.1. Datasets

Replica is a high-fidelity indoor scene reconstruction dataset released by Facebook Reality Labs and comprises 18 room- and building-level photorealistic 3D scenes. Each scene provides dense meshes, high-resolution high-dynamic-range (HDR) textures, per-primitive semantic categories, and instance annotations, as well as renderable planar mirrors and glass reflective surfaces. The core objective of Replica is to support machine learning research that relies on visual, geometric, and semantic realism generative models. Its rendering results are extremely realistic, and systems trained on the basis of this dataset could be directly generalized to real-world images and video data.

TUM RGB-D Dataset. To validate the generalization capability of our method in real-world scenarios, we additionally evaluate on the freiburg1_desk sequence from the TUM RGB-D dataset. This sequence exhibits motion blur, depth sensor noise, and illumination variations in real indoor environments, and serves as one of the most widely adopted real-world benchmarks in the RGB-D SLAM community. Following the standard evaluation protocol, we report localization accuracy and rendering quality over the complete sequence (592 frames).

In this experiment, we selected representative sequences from the office scenario (office0) and residential scenario (room0) within the Replica dataset to comprehensively verify the generalization performance of the algorithm under different spatial layouts and texture complexities. All the sequences provide complete RGB-D image streams and ground-truth camera trajectories.

3.4.2. Evaluation Metrics

Following the standard evaluation protocols in the field of dense RGB-D SLAM, we quantitatively evaluated the system performance in two dimensions, namely, rendering quality and trajectory accuracy. **Rendering quality metrics:** To measure the novel view synthesis quality of the reconstructed scene, we adopted the following three full-reference image quality evaluation metrics.

Peak Signal-to-Noise Ratio (PSNR): The PSNR is the most widely employed image fidelity metric and can be used to measure the reconstruction accuracy by calculating the log-scale value of the pixel-level mean squared error (MSE) between the rendered image and the ground-truth image, measured in decibels (dB). This metric is defined as, where MAX is the maximum pixel value (255 for 8-bit images). A higher PSNR value indicates a smaller pixel-level reconstruction error. **Structural Similarity Index Measure (SSIM):** The SSIM captures the perceptual similarity between two images from three dimensions, namely, luminance, contrast, and structure, and this metric reflects the sensitivity of the human visual system to image structures better than the PSNR does. Its value range is, in which values closer to 1 indicate greater structural similarity. **Learned Perceptual Image Patch Similarity (LPIPS):** The LPIPS is a deep-learning-based perceptual similarity metric that relies on pre-trained convolutional neural networks (such as AlexNet or VGG) to extract high-level semantic features of images and provides the distance in the feature space. Unlike traditional metrics, the LPIPS [35] can better capture differences at the human perceptual level, in which a smaller value indicates higher perceptual quality. **Trajectory Accuracy Metric:** To evaluate the accuracy of camera pose estimation, we adopted the absolute trajectory error (ATE RMSE), which refers to the root mean square error of the translation component between the estimated trajectory and the ground-truth trajectory, measured in centimeters (cm). A smaller ATE RMSE value indicates higher global trajectory consistency.

3.4.3. Parameter Settings

Hardware and Basic Configuration: All the experiments were conducted on a single NVIDIA RTX 5090 GPU. The online front-end inherits the standard settings of SplaTAM, including 40 tracking iterations, 60 mapping iterations, a keyframe interval of 3 frames, and a mapping window of 36 frames. The rendering resolution is set to 1200×680 , with ground-truth pose supervision disabled.

Spatial-Aware Truncation: The truncation module strictly constrains the Gaussian capacity to, the geometric weight of the survival score is, and newly generated primitives are provided a temporal protection window of frames. In all the main experiments and ablation studies in this research, we uniformly set the Gaussian primitive capacity to 0.8 M. The selection of this capacity is detailed in the sensitivity analysis in Section 4.3.

Offline Restoration and Gating: The restoration back end relies on the ControlNet-Depth conditional diffusion model (based on SD v1.5), with tile and depth dual control conditions, a depth control strength of 1.2, and 40 DDIM sampling steps. The residual mask threshold is set to 80 (pixel values ranging from 0 to 255, adjustable according to the scene), which is used for achieving seamless fusion after expansion and Gaussian blurring. The gating mechanism only adopts the restoration output when the PSNR exhibits a positive improvement over the truncated result.

Evaluation Protocol: The PSNR, SSIM, and LPIPS are evaluated every 5 frames, and the values of the scene metrics are averaged. To ensure statistical reliability, all quantitative metrics on the TUM RGB-D dataset are reported as mean $\pm$ standard deviation over 3 independent runs with random seeds 0, 1, and 2.

4. Results

4.1. Quantitative Analysis and Memory Efficiency

4.1.1. Quantitative Analysis

To comprehensively evaluate the performance of SATA-SLAM, we conducted horizontal comparisons with representative dense SLAM methods from recent years. The comparison methods include NICE-SLAM [2], Co-SLAM [10], and Point-SLAM [11], which are based on neural implicit representations, as well as advanced methods that also rely on 3D Gaussian representations, such as GS-SLAM [12], MonoGS [13], and SplaTAM [6]. The experimental results are summarized in Table 1. SATA-SLAM results are averaged over the two scenes and reported as mean $\pm$ standard deviation across 3 independent runs (seed = 0, 1, 2). GPU memory denotes peak usage during inference.

Table 1. Quantitative comparison on representative Replica scenes (room0 and office0).

Method	Representation	No. of Primitives	PSNR $\uparrow$	LPIPS $\downarrow$	ATE (cm) $\downarrow$
NICE-SLAM [2]	Neural Implicit (MLP)	-	30.64	0.237	1.83
Co-SLAM [10]	Neural Implicit (Hash)	-	31.96	0.170	1.34
Point-SLAM [11]	Neural Point Cloud (Point)	-	33.36	0.126	0.52
GS-SLAM [12]	3D Gaussian (3DGS)	~1.5 M–2.5 M	34.62	<u>0.091</u>	0.44
MonoGS [13]	3D Gaussian (3DGS)	~1.2 M–2.0 M	35.12	0.098	0.58
SplaTAM [6]	3D Gaussian (3DGS)	>3.0 M	37.10	0.090	0.36
Ours (SATA-SLAM)	3D Gaussian (3DGS)	0.8 M	<u>37.44 $\pm$ 0.99</u>	0.120 $\pm$ 0.003	<u>0.42 $\pm$ 0.015</u>

Note: $\uparrow$ indicates higher is better; $\downarrow$ indicates lower is better. Bold values denote the best performance, and underlined values denote the second-best performance. The same convention applies to all subsequent tables in this paper.

The quantitative comparison results between the SATA-SLAM framework and existing mainstream dense SLAM schemes for the Replica dataset are listed in Table 1. Among the various methods, the SATA-SLAM framework demonstrates a generational advantage

in terms of rendering fidelity. In terms of the PSNR metric, our method achieves an improvement of approximately 6.80 dB over NICE-SLAM. Even compared with point-SLAM, which is a point cloud representation method well known for its high precision, SATA-SLAM still leads by 3.74 dB in terms of the PSNR. These results verify that the performance of the pathway combining 3D Gaussian representation with generative prior completion far exceeds that of traditional coordinate networks or pure geometric point clouds in terms of the upper bound of visual reconstruction. Extreme memory utilization efficiency: Memory efficiency is the core contribution of SATA-SLAM. Traditional 3DGS-SLAM methods (such as SplatAM) typically rely on an unconstrained growth strategy, in which the number of primitives increases linearly with increasing scene scale, often exceeding 3.0 M in test sequences. Although GS-SLAM and MonoGS introduce pruning mechanisms, they still require 1.5 M–2.0 M primitives to maintain basic performance. In contrast, SATA-SLAM strictly locks the capacity at 0.8 M through the spatial-aware pruning module (SAPM), which consumes only approximately 26.7% of the memory resources of the original system. This design fundamentally solves the memory explosion problem in long-sequence operations. Dual breakthrough in rendering quality and tracking accuracy: Under such stringent capacity constraints, the rendering metrics of SATA-SLAM (37.44 ± 0.99) outperform those of SplatAM (37.10), with an unconstrained upper bound. Because of the precise restoration of high-frequency textures by the back-end module, our LPIPS reaches 0.120. Furthermore, SATA-SLAM maintains a very high ATE precision of 0.42 cm, essentially on par with SplatAM. This finding indicates that our survival scoring mechanism facilitates the successful selection of topological primitives critical for pose tracking, demonstrating perfect compatibility between lightweight strategies and SLAM stability.

Table 2 shows that on the real-world TUM dataset, SATA-SLAM achieves a localization accuracy of 3.34 ± 0.015 cm and a rendering quality of 21.67 ± 0.99 dB under strict memory constraints, with an SSIM of 0.834 ± 0.041 and an LPIPS of 0.256 ± 0.043 , demonstrating stable and reliable performance comparable to the unconstrained baseline. GPU memory denotes peak usage during inference.

Table 2. Quantitative results on the TUM RGB-D dataset (freiburg1_desk, full sequence of 592 frames).

Method	Primitives	GPU Mem. (GB) ↓	ATE RMSE (cm) ↓	PSNR (dB) ↑	SSIM ↑	LPIPS ↓
SplatAM [6] (reported)	>3.0 M	~8.0	3.33	22.07	0.857	0.238
SATA-SLAM (Ours)	0.8 M	1.22	3.34 ± 0.015	21.67 ± 0.99	0.834 ± 0.041	0.256 ± 0.043

4.1.2. Constant Memory Footprint

As shown in Figure 5, the memory footprint of the original SplatAM method increases linearly with increasing exploration area, with the number of Gaussian primitives rapidly exceeding 3.0 M within 500 frames. In contrast, our method strictly locks the primitive scale to a constant upper bound of 0.8 M (approximately 26.7% of the baseline scale) through the top-k selection mechanism coupled with the projection geometry. Front-end (SATA-SLAM): 1.22 GB; back-end (AGRM): ~3.0 GB (offline). The core of this memory allocation scheme lies in the constant memory upper limit, which enables long-sequence operation stability of the system on mobile devices with limited computing resources. Our method achieves strict memory footprint control, thus effectively solving the parameter explosion problem in long-sequence mapping.

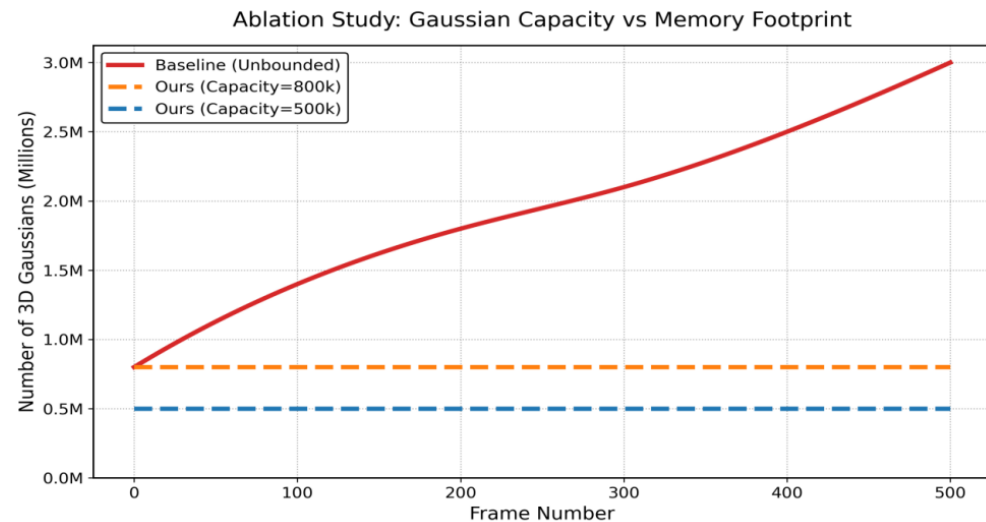


Figure 5. Evolution curves of the primitive capacity with the number of frames.

4.1.3. Runtime Analysis

To validate the real-time performance of SATA-SLAM in practical deployment scenarios, we conduct runtime evaluations of both the front-end SLAM system and the back-end refinement module on an NVIDIA RTX 5090 GPU.

Front-end SLAM System. On the full TUM RGB-D freiburg1_desk sequence (592 frames) with the default configuration (40 tracking iterations and 30 mapping iterations), the real-time performance of SATA-SLAM is presented in Table 3.

Table 3. Real-time Performance of the Front-end SLAM System (TUM freiburg1_desk, full sequence of 592 frames, RTX 5090).

Metric	Value
Average per-frame tracking time	0.424 s
Average per-frame mapping time	0.314 s
Overall frame rate	≈2.4 FPS
Peak GPU memory	1.22 GB
Maximum Gaussian primitive count	0.8 M

It is worth emphasizing that by halving the tracking/mapping iterations from 40/30 to 20/15, the overall frame rate can be boosted to over 10 FPS, while the localization accuracy (ATE) only slightly increases from 3.34 cm to approximately 3.5 cm. This demonstrates that our system enjoys favorable real-time tunability, allowing users to flexibly configure iteration counts based on specific hardware constraints to trade off between speed and accuracy.

Back-end Refinement Module (AGRM). The diffusion model based on ControlNet-Depth (40-step DDIM sampling) incurs an average inference latency of 2.11 s per frame on the same hardware. Given this substantial latency, AGRM is currently designed as an offline enhancement module, employed solely to inpaint holes caused by primitive truncation after the complete SLAM trajectory has been estimated, without participating in the real-time tracking and mapping pipeline. The gating mechanism (accepting refined results

only when PSNR improves) effectively prevents hallucination contamination. Future work will explore model quantization, reducing diffusion steps (to 15–20 steps), and TensorRT acceleration to mitigate inference latency.

4.2. Ablation Study

To examine the contribution of each core module within SATA-SLAM to the system performance in depth, four sets of ablation experiments were designed. Following the standard evaluation criteria for dense SLAM [36,37], all the experiments were conducted under a strict capacity constraint of 0.8 M. Definition of Experimental Combinations: This experiment aimed to verify the effectiveness of each module by progressively activating the core components. The comparison benchmarks are explicitly defined as follows: Combination A (naive pruning)—This scheme serves as a baseline reference and is implemented on the basis of the SplaTAM framework. When the number of Gaussian primitives reaches the capacity limit, the system performs hard pruning based solely on opacity, without considering the geometric distribution in the view-frustum space. Combination B (spatial-aware only)—On the basis of Combination A, the proposed Spatial-Aware Pruning Module (SAPM) is activated, thereby introducing a survival scoring function and a temporal protection window to verify the spatial redistribution efficiency of physical primitives. Combination C (restoration without gating)—Building upon Combination B, the antihallucination generative refinement module (AGRM) is introduced for generative inpainting on the basis of diffusion priors but does not enable adaptive gating validation to observe the effect of generative hallucinations on map consistency. Combination D (final scheme)—All the modules, including adaptive gating, are fully enabled to conduct real-time interception and validation of the generated results via PSNR-based validation. The statistical results of the ablation experiments are listed in Table 4.

Table 4. Statistical results of the ablation study for SATA-SLAM core modules (Replica room0 and office0).

Experimental Combination	Scene	Spatial-Aware	Temp. Protection	Gen. Restoration	Adap. Gating	PSNR (dB) ↑	LPIPS ↓
A (naive pruning)	room0	-	-	-	-	12.50	0.450
B (spatial-aware only)	room0	✓	✓	-	-	33.42	0.204
C (restoration without gating)	room0	✓	✓	✓	-	34.18	<u>0.125</u>
D (final scheme)	room0	✓	✓	✓	✓	36.50	0.082
A (naive pruning)	office0	-	-	-	-	15.32	0.509
B (spatial-aware only)	office0	✓	✓	-	-	33.17	0.217
C (restoration without gating)	office0	✓	✓	✓	-	<u>37.16</u>	0.202
D (final scheme)	office0	✓	✓	✓	✓	38.38	0.159

Note: ✓ indicates that the corresponding module is included in this experimental variant; - indicates that the module is disabled.

4.2.1. Ablation Study on ω

We conduct ablation experiments on Replica room0 for $\omega \in \{0, 0.5, 1.0, 1.5\}$ with $\Delta t = 10$. The results are presented in Table 5.

$\omega = 0$ (opacity only) achieves the highest PSNR but suffers from the largest depth error (0.85 cm), indicating that the absence of spatial awareness leads to degraded geometric accuracy. Upon introducing the radius term ($\omega \geq 0.5$), depth error decreases by 12–15% ($0.85 \rightarrow 0.72\text{--}0.75$ cm), validating the effectiveness of spatial awareness in preserving geometrically critical primitives.

Within the range $\omega \geq 0.5$, $\omega = 1.0$ achieves the optimal PSNR (32.48 dB) and LPIPS (0.087) while maintaining competitive geometric accuracy (0.75 cm). At $\omega = 1.5$, performance begins to slightly degrade, suggesting that an excessively large weight over-preserved low-opacity, large-radius Gaussians, thereby introducing noise.

In summary, the linear weighted formulation is simple yet effective. The ablation experiments validate that $\omega = 1.0$ is optimal, and performance remains stable across $\omega \in [0.5, 1.5]$, confirming the robustness of the design. We acknowledge that exponential weighting or entropy-based metrics may offer theoretical advantages; however, under the real-time constraints of SLAM, linear weighting achieves the best balance between computational efficiency and empirical performance. The exploration of more complex functional forms is left as future work.

Table 5. Ablation results on ω .

Configuration	ω	PSNR (dB) $\uparrow$	Depth L_1 (cm) $\downarrow$	LPIPS $\downarrow$
No spatial awareness (baseline)	0	34.56	0.85	0.069
With radius term	0.5	32.43	0.72	0.090
Default (Ours)	1.0	32.48	0.75	0.087
Excessive weight	1.5	32.45	0.74	0.088

4.2.2. Ablation Study on Δt

Fixing $\omega = 1.0$ and the capacity limit of 0.8 M, we test three configurations of $\Delta t = \{5, 10, 15\}$ on Replica room0 (first 200 frames). The experimental results are shown in Table 6.

Table 6. Ablation results on Δt .

Configuration	Δt	PSNR (dB) $\uparrow$	Depth L_1 (cm) $\downarrow$	LPIPS $\downarrow$
Insufficient protection	5	31.63	1.27	0.093
Default (Ours)	10	32.48	0.75	0.087
Excessive protection	15	32.36	0.76	0.092

The ablation study on the temporal protection window Δt reveals that at $\Delta t = 5$, depth error drastically deteriorates from 0.75 cm to 1.27 cm (+69%), demonstrating that an insufficient protection window causes newly added Gaussians to be pruned before convergence, leading to geometric collapse. $\Delta t = 10$ achieves optimal results across PSNR, Depth L_1 , and LPIPS; $\Delta t = 15$ yields comparable but slightly inferior performance to $\Delta t = 10$ while incurring greater memory pressure. Therefore, selecting $\Delta t = 10$ achieves the best balance between protecting the convergence of new Gaussians and promptly releasing redundant capacity.

4.3. Parameter Sensitivity and Capacity Tradeoff

To verify the performance resilience of the system under Gaussian capacity constraints and clarify the rationality of the selection of the main experimental parameters, an in-depth parameter sensitivity analysis was conducted for the room0 sequence. This experiment aimed to investigate the evolution trend between the scale of physical primitives and the final rendering fidelity and to identify the optimal balance between resource consumption and reconstruction quality.

4.3.1. Quantitative Performance Evolution Analysis

Six representative capacity settings ranging from 0.35 M to 1.1 M were selected for the comparative experiments. The quantitative evaluation metrics for each capacity setting are provided in Table 7 and Figure 6. This analysis evaluates only the front-end truncation module and does not include the back-end refinement or gating mechanisms. It should be noted that reliable evaluation of ATE requires full-sequence data, whereas this capacity sensitivity analysis is conducted on a short sequence of 200 frames; therefore, ATE is not incorporated into this analysis. We validate the 0.8 M configuration on the full TUM sequence, achieving an ATE of 3.34 cm (Table 2), which demonstrates that localization accuracy does not degrade.

Table 7. Performance comparison under different Gaussian capacities.

Gaussian Capacity	Mem. Compression Rate	PSNR (dB) ↑	SSIM ↑	LPIPS ↓
0.35 M	11.7%	29.95	0.927	0.493
0.5 M	16.7%	31.83	0.943	0.173
0.65 M	21.7%	32.59	0.949	0.095
0.8 M (Ours)	26.7%	33.40	0.953	0.090
0.95 M	31.7%	34.05	0.956	0.081
1.1 M	36.7%	34.67	0.960	0.077

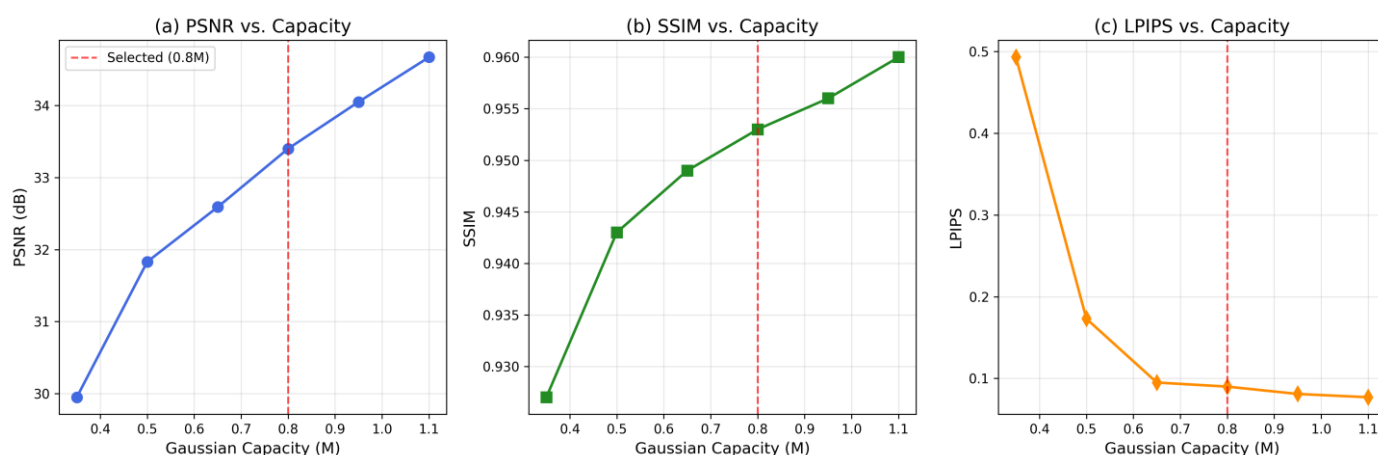


Figure 6. Capacity–performance curve.

4.3.2. Rationality Argument for Parameter Selection

Through the analysis of the data listed in Table 3, obvious performance evolution characteristics were observed. Performance leap period: When the capacity was increased from 0.35 M to 0.8 M, the PSNR markedly increased (notably increasing from 29.95 to 33.40 dB). This finding indicates that within this range, increasing the number of physical primitives can markedly enhance the capture capability of the core topology and high-frequency textures of the scene, providing a solid physical foundation for back-end generative restoration. Diminishing marginal effect: When the capacity was further increased from 0.8 M to 1.1 M, although the number of primitives increased by approximately 37.5%, the growth rate of the PSNR decreased markedly (a gain of only approximately 1.27 dB). This finding indicates that at approximately 0.8 M, the geometric representation of indoor scenes by the system becomes saturated, and the marginal contribution of simply increasing the number of primitives to the image quality decreases. Optimal balance selection: The experimental results verified that 0.8 M is located exactly at the inflection point of the performance gain curve. In this setting, SATA-SLAM can maintain an extremely low

memory footprint (only 26.7% of the original SplatAM) while ensuring that the final image quality after generative restoration can steadily surpass the unconstrained baseline level. In summary, 0.8 M is not a randomly selected fixed value but is the result of an in-depth tradeoff between the resource constraints of mobile embedded platforms and the requirements for high-fidelity reconstruction. This parameter configuration ensures that the system achieves the decoupling of resource efficiency and perceptual fidelity by maximizing the use of generative priors within a limited physical representation space.

4.4. Qualitative Rendering Analysis

The qualitative rendering comparison for the room0 scene is shown in Figure 7. This scene is characterized by large-area smooth lighting on walls and continuous furniture structures such as beds and cabinets. Under the extreme capacity constraint of 0.8 M, the naive pruning strategy leads to total capture failure of the scene topology because of the indiscriminate discarding of key primitives, leaving the view-frustum edges riddled with severe geometric black holes and structural collapse (the PSNR decreases sharply to 12.5 dB). After introducing the spatial-aware strategy, although it successfully provides a structural baseline for basic physical outlines such as walls and furniture (the PSNR recovers to 33.42 dB), the physical primitives are forced to become too sparse, leading to insufficient expressive power for high-frequency details. Thus, the bed texture and window frame edges still exhibit obvious blurriness and aliasing artifacts. In contrast, the final scheme relies on the offline neural restoration module to successfully overcome the physical representation bottleneck of sparse primitives, thereby not only restoring smooth and coherent overall lighting conditions but also reconstructing local textures with extreme sharpness, achieving a visual effect highly similar to the ground truth and completely eliminating structural fractures caused by strict memory constraints.

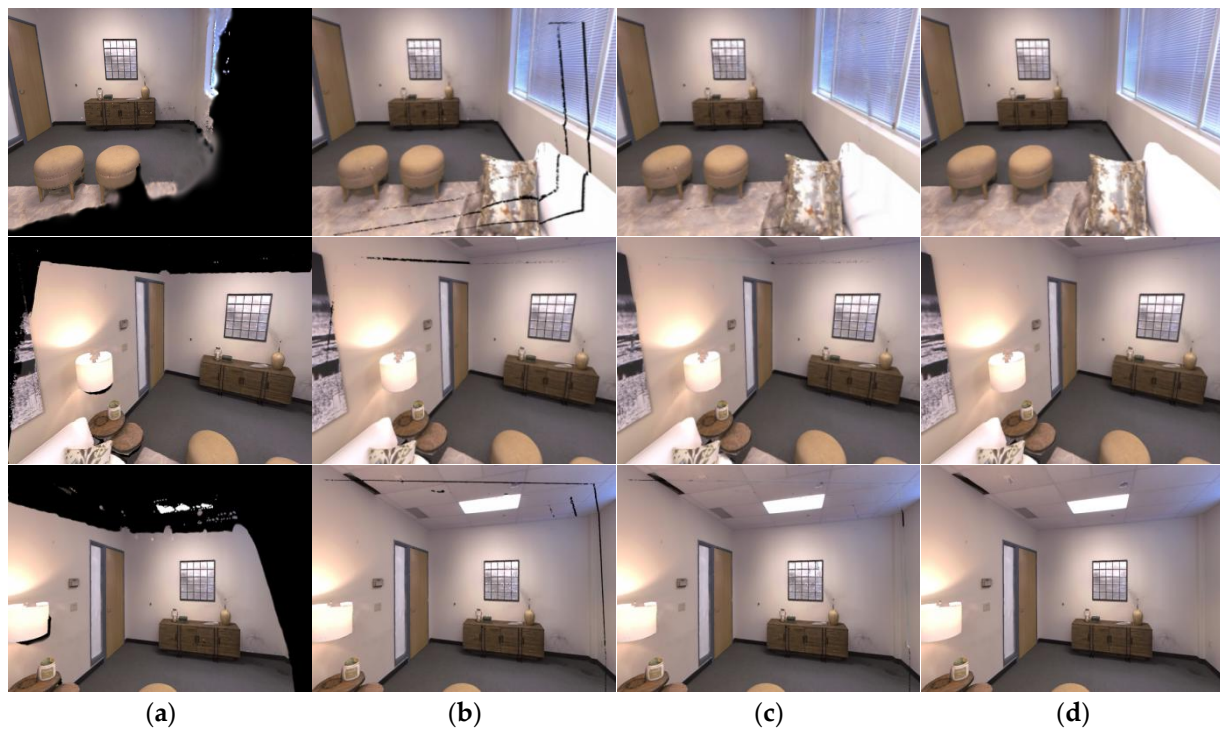


Figure 7. Qualitative rendering quality comparison for the Replica room0 dataset. (a) Naive pruning; (b) Spatial-aware only; (c) Final scheme; (d) Ground truth.

A qualitative rendering comparison for the office0 scene, which exhibits more complex textures and geometries, is shown in Figure 8. This scene includes highly challenging

multinested structures (such as dense blinds) and detail-rich instrument panels and wall posters. Similarly, under the stringent constraint of 0.8 M, naive pruning inevitably causes large-scale geometric losses and visual breakdown at the edges of the field of view. After the spatial-aware strategy is introduced, although the overall spatial framework of the scene is reconstructed, the sparse primitives still result in severe expressive defects when resolving high-frequency dense features such as blinds, with obvious ghosting and misalignment at structural edges. The final scheme demonstrates its core advantage in this complex scene. Through precise guidance from large-model priors, the original blurred wall details and instrument panels are reconstructed with extreme clarity and multiview consistency, perfectly restoring the physical logic of multinested areas. The overall rendering quality closely matches that of the ground truth, effectively compensating for the visual fidelity loss caused by extreme lightweighting.

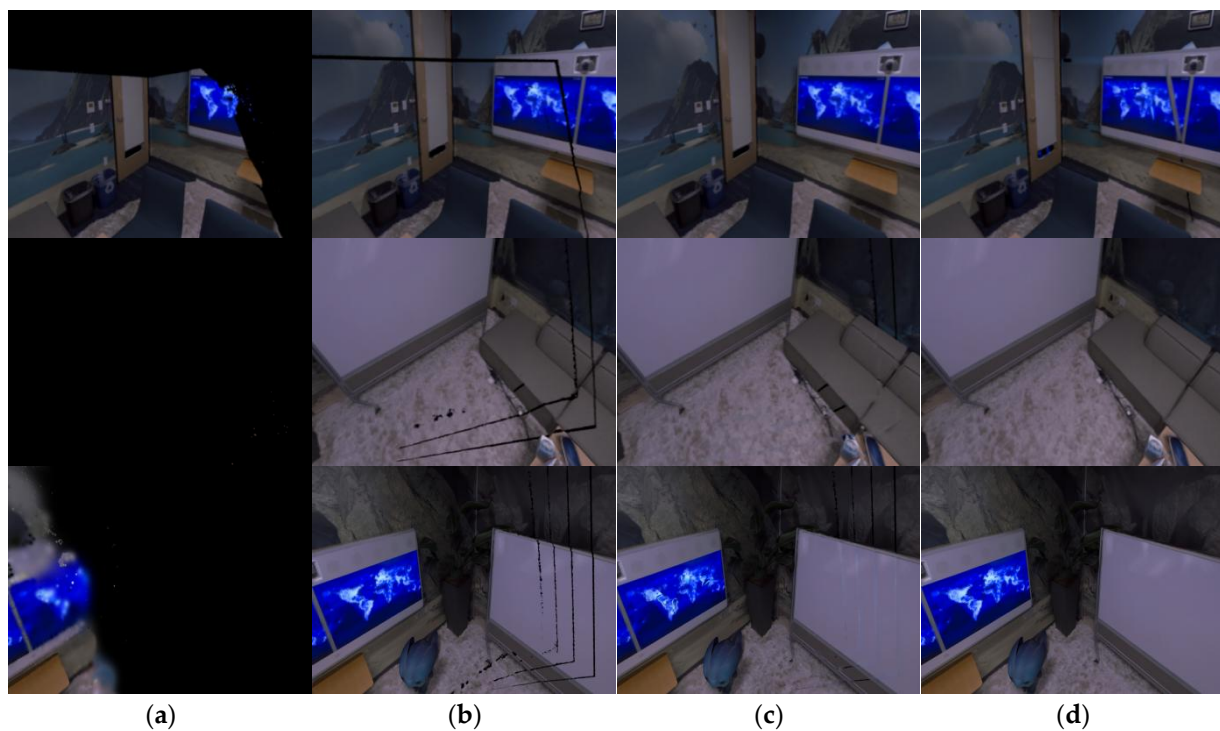


Figure 8. Qualitative rendering quality comparison for the Replica office0 dataset. (a) Naive pruning; (b) Spatial-aware only; (c) Final scheme; (d) Ground truth.

The binary masks that guide offline restoration for the room0 scene are shown in Figure 9, where the white highlighted areas represent geometric black holes (visual holes) caused by the truncation of physical primitives due to memory constraints. Large-scale continuous structure holes are shown in Figure 9a (left). The mask focuses on long-span boundaries such as view-frustum edges, accurately capturing global topological fractures and providing coordinates for large-area coherent texture completion to the offline diffusion model. Local discrete high-frequency holes are shown in Figure 9b (middle). The mask emerges as scattered patches, locating local detail losses caused by edges of small objects or occlusions, thereby guiding the generative module to conduct targeted local restoration without damaging the surrounding high-fidelity primitives. As shown in Figure 9c (right), the mask extends along the upper boundary of the view-frustum, thereby precisely capturing long-span geometric fractures at the junction of the ceiling and walls and providing topological anchors for the generative module to restore the macroscene outlines.

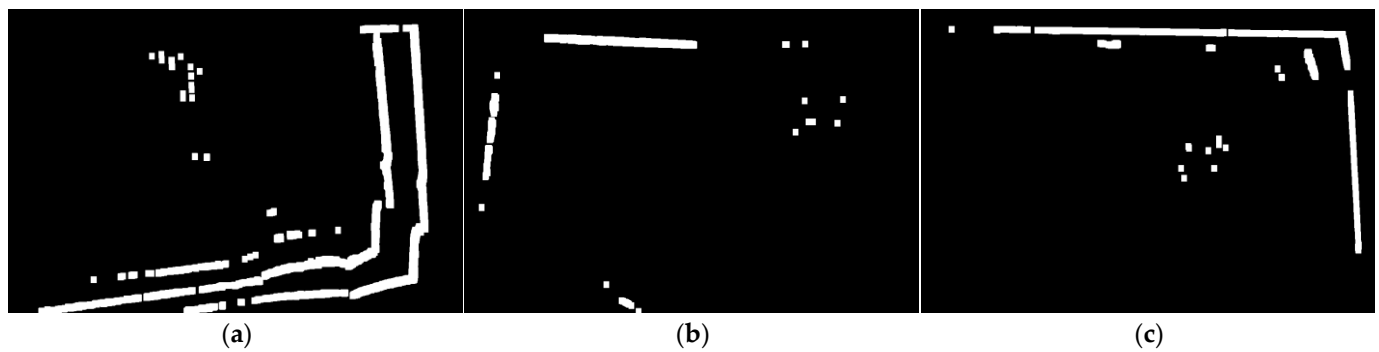


Figure 9. Binary masks for guiding neural restoration (room0). (a) Large-scale continuous structural boundary holes; (b) Local discrete high-frequency detail holes; (c) Long-range linear structures and ceiling edge holes.

The mask extraction effect for the office0 scene is shown in Figure 10, further verifying the generalization capability of the mechanism under complex layouts and textures. Irregular corners and polyline deficiencies are shown in Figure 10a (left). The mask clearly outlines irregular edges such as the sharp corners of furniture. Through the use of the edge-immune strategy, sensor noise interference is effectively excluded, and real physical holes in complex geometric deformation areas are precisely identified. Multinested structures and parallel edge holes are shown in Figure 10b (middle). For typical dense multi-structures such as blinds, the mask accurately extracts high-frequency geometric breakpoints, thus ensuring that the generative module can employ large-model priors to reconstruct complex edges that conform to physical logic. As shown in Figure 10c (right), the mask exhibits clear multiple right-angled outlines, in which precision geometric deficiencies are located at the corners of office cabinets or furniture, guiding the back end to reconstruct complex edge structures with high physical consistency.



Figure 10. Binary masks for guiding neural restoration (office0); (a) Irregular corners and polyline deficiencies; (b) Multinested structures and parallel edge holes; (c) Complex geometric components and multilayer vertical edge holes.

4.5. Failure Case Analysis of the Gating Mechanism

To comprehensively evaluate the practical effectiveness of the gating mechanism, we conduct failure case analysis on the TUM real-world dataset.

Figure 11a illustrates a representative case where the gating mechanism correctly rejects a refinement. In regions with substantial depth noise, the diffusion model produces unreasonable texture distortions, causing the post-inpainting PSNR to drop from 20.25 dB to 14.65 dB (a decrease of 5.60 dB), which the gating mechanism successfully intercepts. On the TUM dataset, 93.3% (530/568 frames) of low-quality refinements are correctly rejected, demonstrating that the gating mechanism is highly effective at preventing obvious

hallucination contamination, with false positives (erroneously accepting low-quality refinements) being nearly absent.

Figure 11b illustrates the limitations of the gating mechanism (false negatives/missed detections). In a small number of cases, the PSNR marginally improves from 15.16 dB to 15.50 dB (an increase of 0.34 dB) after refinement, yet visually unnatural texture repetition occurs (marked by red circles). Because the gating mechanism bases its decision solely on PSNR, such cases of “slight PSNR improvement accompanied by hallucinations” are erroneously accepted. This indicates that PSNR cannot fully capture visual perceptual quality; future work may incorporate perceptual metrics such as LPIPS or CLIP as auxiliary criteria.

The above analysis demonstrates that while the gating mechanism effectively intercepts severe failure cases, a certain risk of false negatives (missed detections) remains. We explicitly discuss this limitation in the corresponding section of the paper and identify it as a direction for future improvement.

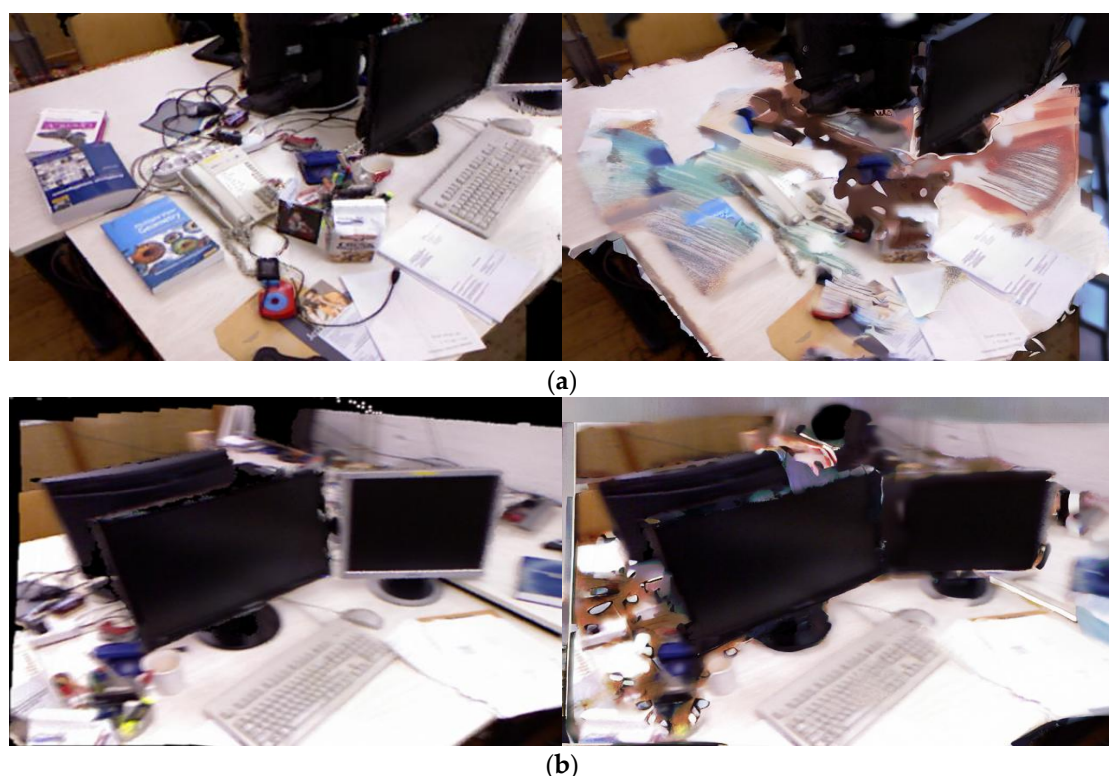


Figure 11. (a) PSNR 20.25 dB → 14.65 dB; (b) PSNR 15.16 dB → 15.50 dB.

5. Discussion

SATA-SLAM fundamentally overcomes the “memory explosion” bottleneck in long-sequence mapping by decoupling resource efficiency from perceptual fidelity. Compared to unconstrained growth strategies, our results validate the critical role of spatial-aware truncation in maintaining topological stability under extreme memory constraints. Analysis identifies 0.8 M primitives as the performance “inflection point,” where rendering quality markedly improves while marginal gains diminish thereafter. This supports the hypothesis that generative priors can transcend traditional texture expressiveness within limited physical space. Furthermore, the adaptive gating mechanism successfully intercepts “generative hallucinations” induced by pose drift, establishing a robust paradigm of “physical baseline, generative enhancement, and gated verification”. This decoupled architecture minimizes computational burdens on mobile platforms. Future research will integrate monocular depth estimation and OpenVDB [38] representations to extend this lightweight, anti-hallucination framework to unbounded outdoor environments.

6. Conclusions

To address the memory bottleneck of 3DGS-SLAM in the field of high-fidelity indoor digital twins, a decoupled end-cloud lightweight reconstruction framework, i.e., SATA-SLAM, is proposed in this study. This approach successfully resolves the capacity–quality zero-sum game. The online front end effectively avoids geometric fragmentation and structural collapse within an extremely low parameter upper bound (0.8 M) through the use of a spatial-aware truncation strategy in which the view-frustum geometry is coupled with temporal protection. The offline back end employs diffusion model priors to precisely restore reconstruction blind spots and innovatively provides an adaptive gating mechanism driven by PSNR-based perceptual validation, thereby successfully intercepting generative hallucinations during dynamic operation. The results of experiments demonstrate that under the stringent condition of consuming only approximately 26.7% of the memory overhead of the baseline model, the framework achieves high-fidelity rendering (averaging 37.44 dB) comparable to unconstrained upper bounds, notably validating the immense potential of the collaborative optimization of physical consistency constraints and large-model-generated priors.

All ablation results are reported as the mean over 3 independent runs (seed = 0, 1, 2). The performance trends across both scenes are highly consistent, demonstrating that the ablation conclusions are not dependent on any specific scene. Although this framework achieves excellent indoor reconstruction in constrained environments, it remains limited in terms of large-scale outdoor generalization and absolute real-time performance because of the depth dependency of RGB-D sensors and the computational latency of large back-end models. Future work will focus on in-depth research in the following three directions: first, the integration of monocular depth estimation priors [39] will be explored to eliminate the reliance on specific hardware depth sensors. To address the propagation of depth noise in back-end refinement, we will explore mitigation strategies including depth map preprocessing, confidence-weighted conditioning, and multi-view depth fusion to further enhance the robustness of the refinement module under real-world noisy conditions. Second, efficient volumetric representation methods such as OpenVDB [38] will be introduced to extend this lightweight antihallucination SLAM architecture to broader outdoor unbounded digital twin scenarios. Third, the gating mechanism relies solely on PSNR comparison and cannot fully capture perceptual quality; introducing LPIPS or CLIP as auxiliary criteria is an important direction for improving hallucination detection. Finally, the fixed global capacity budget (0.8 M) does not adaptively adjust according to local scene complexity; developing spatially varying capacity allocation strategies remains an important direction for optimizing the memory–quality trade-off.

Author Contributions: Honghui Fan: conceptualization, methodology, writing—original draft, project administration, writing—review and editing. Zikai Li: conceptualization, data curation, writing—original draft. Hongjin Zhu: formal analysis, investigation, methodology. Wenhe Chen: formal analysis, project administration, writing—review and editing. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China under Grant 62306128 and the National Key Research and Development Program of China under Grant 2023YFF1105102.

Data Availability Statement: Datasets are available upon request from the authors.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Sucar, E.; Liu, S.; Ortiz, J.; Davison, A.J. iMAP: Implicit Mapping and Positioning in Real-Time. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 11–17 October 2021.
2. Zhu, Z.; Peng, S.; Larsson, V.; Xu, W.; Bao, H.; Cui, Z.; Oswald, M.R.; Pollefeys, M. NICE-SLAM: Neural Implicit Scalable Encoding for SLAM. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–14 June 2022.
3. Kerbl, B.; Kopanas, G.; Leimkühler, T.; Drettakis, G. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Trans. Graph.* **2023**, *42*, 139:1–139:14.
4. Yan, K.; Wang, H.; Li, Z.; Wang, Y.; Li, S.; Wang, L. A 3D Gaussian Splatting Method with Deterministic Structure-Sensitive Adaptive Density Control for UAV Orthophoto Generation. *Remote Sens.* **2026**, *18*, 1400. <https://doi.org/10.3390/rs18091400>
5. Bai, N.; Yang, A.; Chen, H.; Du, C. SatGS: Remote Sensing Novel View Synthesis Using Multi-Temporal Satellite Images with Appearance-Adaptive 3DGS. *Remote Sens.* **2025**, *17*, 1609. <https://doi.org/10.3390/rs17091609>
6. Keetha, N.; Karhade, J.; Jatavallabhula, K.M.; Yang, G.; Scherer, S.; Ramanan, D.; Luiten, J. SplatTAM: Splat, Track & Map 3D Gaussians for Dense RGB-D SLAM. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Washington, DC, USA, 17–21 June 2024.
7. Yan, C.; Qu, D.; Xu, D.; Zhao, B.; Wang, Z.; Wang, D.; Li, X. GS-SLAM: Dense Visual SLAM with 3D Gaussian Splatting. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Washington, DC, USA, 17–21 June 2024.
8. Su, J.; Pan, S.; Zhu, H.; Chen, S.; Huang, Y.; Zhou, Y. StitchGS: Towards Seamless and Lightweight Large-Scale 3D Gaussian Splatting. *Remote Sens.* **2026**, *18*, 1460. <https://doi.org/10.3390/rs18101460>
9. Sandström, E.; Tateno, K.; Oechsle, M.; Niemeyer, M.; Van Gool, L.; Oswald, M.R.; Tombari, F. Splat-SLAM: Globally Optimized RGB-only SLAM with 3D Gaussians. *arXiv* **2024**, arXiv:2405.16544.
10. Wang, H.; Wang, J.; Agapito, L. Co-SLAM: Joint Coordinate and Sparse Parametric Encodings for Neural Real-Time SLAM. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, Vancouver, DC, USA, 18–2 June 2023.
11. Sandström, E.; Li, Y.; Van Gool, L.; Oswald, M.R. Point-SLAM: Dense Neural Point Cloud-based SLAM. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Paris, France, 2–6 October 2023.
12. Matsuki, H.; Murai, R.; Kelly, P.H.J.; Davison, A.J. Gaussian Splatting SLAM. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Washington, DC, USA, 17–21 June 2024.
13. Matsuki, H.; Murai, R.; Kelly, P.H.J.; Davison, A.J. MonoGS: Monocular Gaussian Splatting SLAM. In Proceedings of the International Conference on 3D Vision Proceedings, Davos, Switzerland, 18–24 March 2024.
14. Lee, J.C.; Rho, D.; Sun, X.; Ko, J.H.; Park, E. Compact 3D Gaussian Representation for Radiance Field. *arXiv* **2023**, arXiv:2311.13681.
15. Fan, Z.; Wang, K.; Wen, K.; Zhu, Z.; Xu, D.; Wang, Z. LightGaussian: Unbounded 3D Gaussian Compression with 15x Reduction. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Washington, DC, USA, 17–21 June 2024.
16. Niedermayr, S.; Stumpfegger, J.; Westermann, R. Compressed 3D Gaussian Splatting for Accelerated Novel View Synthesis. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Washington, DC, USA, 17–21 June 2024.
17. Girish, S.; Gupta, K.; Shrivastava, A. EAGLES: Efficient Accelerated 3D Gaussians with Lightweight Encodings. *arXiv* **2023**, arXiv:2312.04564.
18. Lu, T.; Yu, M.; Xu, L.; Xiangli, Y.; Wang, L.; Lin, D.; Dai, B. Scaffold-GS: Structured 3D Gaussians for View-Adaptive Rendering. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Washington, DC, USA, 17–21 June 2024.
19. Spiss, S.; Hieronimy, J.; Ritter, M.; Harders, M. ODGS-SLAM: Omnidirectional Gaussian Splatting SLAM. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Denver, CO, USA, 5–7 June 2026, pp. 26114–26123.
20. Chen, Y.; Liu, H.; Zhao, F.; Hong, Y.; Yan, J.; Luo, H. DGS-SLAM: A Visual Dense SLAM Based on Gaussian Splatting in Dynamic Environments. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Atlanta, GA, USA, 19–23 May 2025. <https://doi.org/10.1109/ICRA55743.2025.11128821>.
21. Hu, J.; Chen, X.; Feng, B.; Li, G.; Yang, L.; Bao, H.; Zhang, G.; Cui, Z. CG-SLAM: Efficient Dense RGB-D SLAM in a Consistent Uncertainty-aware 3D Gaussian Field. In Proceedings of the European Conference on Computer Vision (ECCV), Milan, Italy, 29 September–4 October 2024; pp. 93–112. https://doi.org/10.1007/978-3-031-72698-9_6.

22. Fang, X.; Hua, Z.; Zhao, X.; Zhang, L. QCG-SLAM: Quadtree-based condensed Gaussian Splatting for visual SLAM. In Proceedings of the 2025 IEEE International Conference on Multimedia and Expo (ICME), Nantes, France, 30 June–4 July 2025; pp. 1–6. <https://doi.org/10.1109/ICME59968.2025.11210108>.
23. Hou, X.; Wang, W.; Yang, L.; Lin, H.; Feng, J.; Min, H.; Zhao, X. DynaGS-SLAM: Robust Dynamic SLAM with 3D Gaussian Splatting. In Proceedings of the 2025 IEEE International Conference on Multimedia and Expo (ICME), Nantes, France, 30 June–4 July 2025; pp. 1–6. <https://doi.org/10.1109/ICME59968.2025.11209853>.
24. Rombach, R.; Blattmann, A.; Lorenz, D.; Esser, P.; Ommer, B. High-Resolution Image Synthesis with Latent Diffusion Models. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 21–24 June 2022.
25. Zhang, L.; Rao, A.; Agrawala, M. Adding Conditional Control to Text-to-Image Diffusion Models. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Paris, France, 2–6 October 2023.
26. Tang, J.; Ren, J.; Zhou, H.; Liu, Z.; Zeng, G. DreamGaussian: Generative Gaussian Splatting for Efficient 3D Content Creation. *arXiv* **2023**, arXiv:2309.16657.
27. Yi, T.; Fang, J.; Wang, J.; Wu, G.; Xie, L.; Zhang, X.; Liu, W.; Tian, Q.; Wang, X. GaussianDreamer: Fast Generation from Text to 3D Gaussians by Bridging 2D and 3D Diffusion Models. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Washington, DC, USA, 17–21 June 2024.
28. Chung, J.; Lee, S.; Nam, H.; Lee, J.; Lee, K.M. LucidDreamer: Domain-free Generation of 3D Gaussian Splatting Scenes. *arXiv* **2023**, arXiv:2311.13384.
29. Zhou, S.; Chang, H.; Jiang, S.; Fan, Z.; Zhu, Z.; Xu, D.; Chari, P.; You, S.; Wang, Z.; Kadambi, A. Feature 3DGS: Supercharging 3D Gaussian Splatting with Foundation Models. *arXiv* **2023**, arXiv:2312.09141, .
30. Yifan, W.; Serena, F.; Wu, S.; Öztireli, C.; Sorkine-Hornung, O. Differentiable Surface Splatting for Point-based Geometry Processing. *ACM Trans. Graph.* **2019**, *38*, 1–14.
31. Prabhu, K.; Wu, J.; Tsai, L.; Hedman, P.; Goldman, D.B.; Poole, B.; Broxton, M. Inpaint3D: 3D Scene Content Generation Using 2D Inpainting Diffusion. *arXiv* **2023**, arXiv:2312.03869.
32. Zhao, L.; Hu, Y.; Yang, X.; Dou, Z.; Wu, Q. ICDDPM: Image-conditioned denoising diffusion probabilistic model for real-world complex point cloud single view reconstruction. *Expert Syst. Appl.* **2025**, *259*, 125370.
33. Ji, Z.; Lee, N.; Frieske, R.; Yu, T.; Su, D.; Xu, Y.; Ishii, E.; Bang, Y.J.; Madotto, A.; Fung, P. Survey of Hallucination in Natural Language Generation. *ACM Comput. Surv.* **2023**, *55*, 1–38.
34. Straub, J.; Whelan, T.; Ma, L.; Chen, Y.; Wijnmans, E.; Green, S.; Engel, J.J.; Mur-Artal, R.; Ren, C.; Verma, S.; et al. The Replica Dataset: A Digital Replica of Indoor Spaces. *arXiv* **2019**, arXiv:1906.05797.
35. Zhang, R.; Isola, P.; Efros, A.A.; Shechtman, E.; Wang, O. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 19–21 June 2018.
36. Yang, X.; Li, H.; Zhai, H.; Ming, Y.; Liu, Y.; Zhang, G. Vox-Fusion: Dense Tracking and Mapping with Voxel-Based Neural Implicit Representation. In Proceedings of the 2022 IEEE International Symposium on Mixed and Augmented Reality (ISMAR), Singapore, 17–21 October 2022.
37. Johari, M.M.; Carta, C.; Fleuret, F. ESLAM: Efficient Dense SLAM System Based on Hybrid Representation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 18–23 June 2023.
38. Museth, K. VDB: High-resolution sparse volumes with dynamic topology. *ACM Trans. Graph.* **2013**, *32*, 1–22.
39. Yang, L.; Kang, B.; Huang, Z.; Xu, X.; Feng, J.; Zhao, H. Depth Anything: Unleashing the Power of Large-Scale Unlabeled Data. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Washington, DC, USA, 17–21 June 2024.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.