MIRAGE: METRIC INFERENCE AND RELIABILITY-AWARE GAUSSIAN ESTIMATION FROM FEED-FORWARD VISION

Anonymous authors
Paper under double-blind review

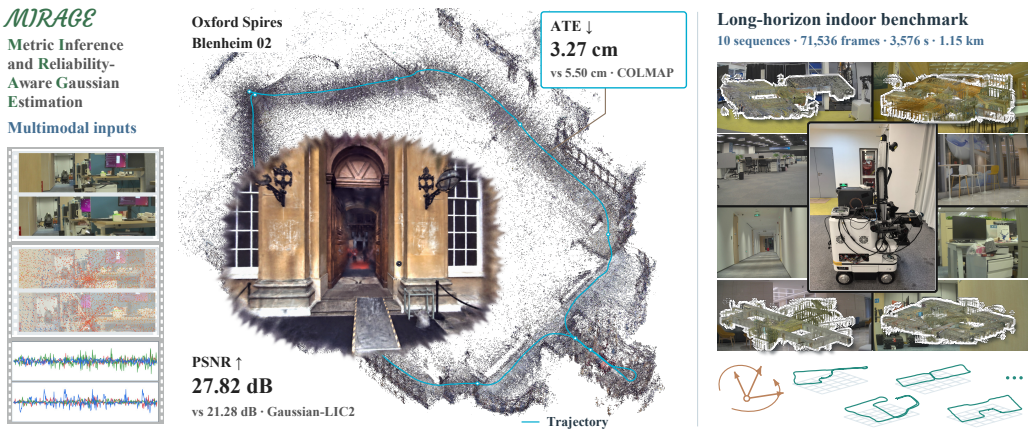

Figure 1: We propose **MIRAGE**, which anchors feed-forward visual geometry to LiDAR–IMU motion through reliability-aware pose refinement. **Left:** on Oxford Spires Blenheim 02, MIRAGE achieves lower ATE than COLMAP (3.27 vs. 5.50 cm) and higher rendering PSNR than Gaussian-LIC2 (27.82 vs. 21.28 dB). **Right:** our synchronized multimodal platform supports a long-horizon indoor benchmark with 10 sequences and MIRAGE-established 6-DoF accurate reference trajectories for evaluating vision-only reconstruction.

ABSTRACT

Feed-forward 3D priors enable high-quality reconstruction from image streams but often lack reliable metric scale and suffer from long-term trajectory drift. In contrast, LiDAR–inertial estimation provides metrically accurate motion but lacks the image-space consistency needed for high-fidelity reconstruction. We propose **MIRAGE**, which anchors feed-forward visual geometry to LiDAR–IMU metric motion and regulates online pose corrections according to correspondence support and cross-modal agreement, jointly enabling accurate metric pose estimation and high-fidelity Gaussian reconstruction. Experiments on Oxford Spires, M2DGR and INS show that MIRAGE achieves accurate metric trajectory estimation, outperforming COLMAP, a widely used but computationally expensive reference for vision-only benchmarks. Motivated by the scarcity of long-horizon monocular benchmarks for large-scale indoor reconstruction, we further build a multimodal data-collection platform and use MIRAGE to establish accurate 6-DoF reference trajectories for a new vision-only benchmark. We benchmark recent monocular large-scale reconstruction methods on this dataset, providing a systematic evaluation of their performance in long-horizon indoor scenes.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

1 INTRODUCTION

Recent advances in 3D scene representation and camera pose estimation have substantially improved the quality and practicality of image-based reconstruction. 3D Gaussian Splatting (3DGS) (Kerbl et al., 2023) represents scenes with optimizable anisotropic Gaussian primitives and an efficient differentiable rasterizer, enabling high-fidelity real-time novel-view synthesis. In parallel, feed-forward geometric models (Leroy et al., 2024; Wang et al., 2025a;b) have made substantial progress in jointly inferring camera poses and scene geometry from unposed images. These advances have further driven the development of increasingly capable online tracking, mapping, and on-the-fly reconstruction systems (Matsuki et al., 2024; Huang et al., 2024; Keetha et al., 2024; Meuleman et al., 2025; Wu et al., 2025). More recent methods combine SLAM frameworks with data-driven feed-forward priors to further improve pose estimation and reconstruction quality in streaming scenarios (Li et al., 2026b; Ren et al., 2026). However, purely visual reconstruction remains subject to global similarity ambiguity, while accumulated errors in learned geometry and pose estimation can lead to scale and motion drift over long sequences. These limitations highlight the need for an explicit metric anchor to achieve both high-fidelity reconstruction and a physically grounded trajectory.

LiDAR–IMU sensing provides the metric scale and motion constraints missing from purely visual reconstruction. However, simply introducing a metric anchor does not resolve how feed-forward visual geometry should influence online pose refinement. Visual priors can improve local alignment and reconstruction quality, yet unreliable corrections may compromise metric consistency, while overly suppressing them can forfeit useful image-space refinement. Since the reliability of visual corrections varies with correspondence support and cross-modal agreement, a fixed fusion strategy is insufficient. The key challenge is therefore to selectively exploit feed-forward visual geometry while preserving a metrically consistent trajectory.

We propose **MIRAGE**[1], an on-the-fly reconstruction framework that combines feed-forward visual geometry with LiDAR–IMU sensing to jointly achieve metrically accurate pose estimation and high-fidelity Gaussian reconstruction. MIRAGE uses LiDAR–IMU motion as a metric anchor and selectively incorporates visual pose corrections according to correspondence support and cross-modal agreement, retaining useful visual refinement without compromising metric consistency. Given the stabilized metric trajectory, MIRAGE further leverages image residuals and LiDAR geometry for structure-aware Gaussian refinement, improving reconstruction quality while keeping the camera poses fixed. Experiments on Oxford Spires (Tao et al., 2026), M2DGR (Yin et al., 2022) and INS (Deng et al., 2025a) demonstrate that MIRAGE achieves accurate and metrically consistent trajectory estimation while maintaining high-fidelity, photorealistic Gaussian reconstruction. Notably, MIRAGE achieves higher trajectory accuracy than COLMAP, despite COLMAP being a computationally expensive method widely used to establish reference trajectories for vision-only benchmarks. This demonstrates MIRAGE's potential to provide reliable reference trajectories for benchmarking vision-only reconstruction methods.

Long-horizon benchmarks for monocular reconstruction in large-scale indoor environments remain limited. To fill this gap, we develop a remotely operated capture platform with hardware-synchronized cameras, LiDAR, and IMU, and collect ten long-horizon sequences across diverse indoor scenes, as shown in Figure 1. MIRAGE is then used to establish accurate 6-DoF reference trajectories for these sequences, forming a vision-only benchmark for large-scale indoor reconstruction. We further evaluate recent long-horizon monocular reconstruction methods on this benchmark.

Our contributions are threefold:

- **Reliability-gated metric trajectory inference.** We anchor feed-forward visual geometry to LiDAR–IMU metric motion and treat visual pose updates as correction proposals, selectively committing them according to correspondence support and cross-modal consistency to improve image-space alignment without compromising metric trajectory accuracy.
- **Pose-fixed structure-aware Gaussian refinement.** We decouple scene refinement from trajectory estimation by fixing the metric camera poses and using image residuals together with LiDAR depth evidence to diagnose and repair oversized, misplaced, and missing primitives, improving geometric detail and appearance without perturbing the recovered trajectory.

[1]The name reflects the gap between visual plausibility and metric consistency.

- **A long-horizon benchmark dataset for real-world indoor reconstruction.** We introduce a large-scale indoor benchmark comprising 10 synchronized RGB–LiDAR–IMU sequences, with 71,536 RGB frames spanning 59.6 minutes and 1.15 km across laboratories, dining areas, offices, and long corridors. The benchmark provides full 6-DoF reference trajectories and supports joint evaluation of long-horizon metric localization and high-fidelity 3D reconstruction.

2 Related Work

2.1 Visual Pose Estimation and Streaming Gaussian Reconstruction

Learned geometry models (Wang et al., 2024; Leroy et al., 2024; Wang et al., 2025a; Lin et al., 2025b) provide dense priors for pose estimation and reconstruction. Recent methods incorporate these priors into visual tracking and global optimization (Murai et al., 2025; Maggio et al., 2026; Maggio & Carlone, 2026), exploit geometric context for reconstruction (Chen et al., 2026; Xie et al., 2026), or combine feed-forward predictions with global structure-from-motion optimization (Deng et al., 2026; Pan et al., 2026).

Gaussian SLAM (Matsuki et al., 2024; Huang et al., 2024; Keetha et al., 2024) extends 3DGS (Kerbl et al., 2023) to incremental tracking and mapping. Other approaches predict Gaussians from image streams (Chen et al., 2024; Li et al., 2025), optimize scenes incrementally (Meuleman et al., 2025; 2026), or combine learned priors with online pose and map optimization (Lin et al., 2025a; Li et al., 2026b; Jiang et al., 2026; Ren et al., 2026). Monocular reconstruction nevertheless remains scale-ambiguous, and learned priors do not guarantee metric accuracy or drift-free trajectories over long traversals.

2.2 LiDAR–Inertial Estimation and Multimodal Gaussian Mapping

LiDAR–inertial estimators (Shan et al., 2020; Xu et al., 2022; Liu et al., 2026) recover metric motion from range and inertial constraints, while visual–LiDAR systems (Shan et al., 2021; Lin & Zhang, 2022; Zheng et al., 2022; 2025; Zhao et al., 2025) incorporate image observations. Multimodal Gaussian mapping (Hong et al., 2024; Lang et al., 2025; Xie et al., 2025; Jiang et al., 2024; Deng et al., 2025b; Cui et al., 2024) uses metric poses and LiDAR geometry for initialization and photometric refinement. Recent methods further incorporate Gaussian-map observations into localization (Hong et al., 2025; Lang et al., 2026) or feed-forward priors into estimation and mapping (Wang et al., 2026; Li et al., 2026a). MIRAGE regulates pose corrections using visual and geometric evidence to support both metric accuracy and rendering quality.

2.3 Benchmarks for Multimodal Localization and Reconstruction

Visual benchmarks (Sturm et al., 2012; Dai et al., 2017; Knapitsch et al., 2017) support tracking or reconstruction evaluation without mobile 3D LiDAR streams. ScanNet++ (Yeshwanth et al., 2023) combines high-resolution RGB, surveyed laser geometry, and mobile RGB-D/IMU data. Mobile datasets vary in sensing and supervision: MulRan (Kim et al., 2020) lacks camera images, Hilti-Oxford (Zhang et al., 2023) uses grayscale cameras, and FAST-LIVO2's self-collected sequences (Zheng et al., 2025) lack pose ground truth. RGB–LiDAR–IMU datasets (Tao et al., 2026; Yin et al., 2022; Jiao et al., 2022; Wei et al., 2025; Zhao et al., 2024) cover diverse environments and platforms, while recent multimodal benchmarks (Deng et al., 2025a; Lang et al., 2026) provide reference geometry and poses for localization and rendering evaluation. Nevertheless, synchronized RGB–LiDAR–IMU benchmarks with full 6-DoF reference trajectories over long, object-rich indoor traversals remain scarce. Our extended captures in laboratories, offices, and dining spaces address this gap, supporting joint evaluation of trajectory accuracy and downstream Gaussian reconstruction.

3 Method

3.1 Overview

Figure 2 summarizes MIRAGE. Given synchronized RGB, LiDAR, and IMU streams, we anchor feed-forward visual geometry to LiDAR–IMU motion for metric camera pose estimation. Corre-

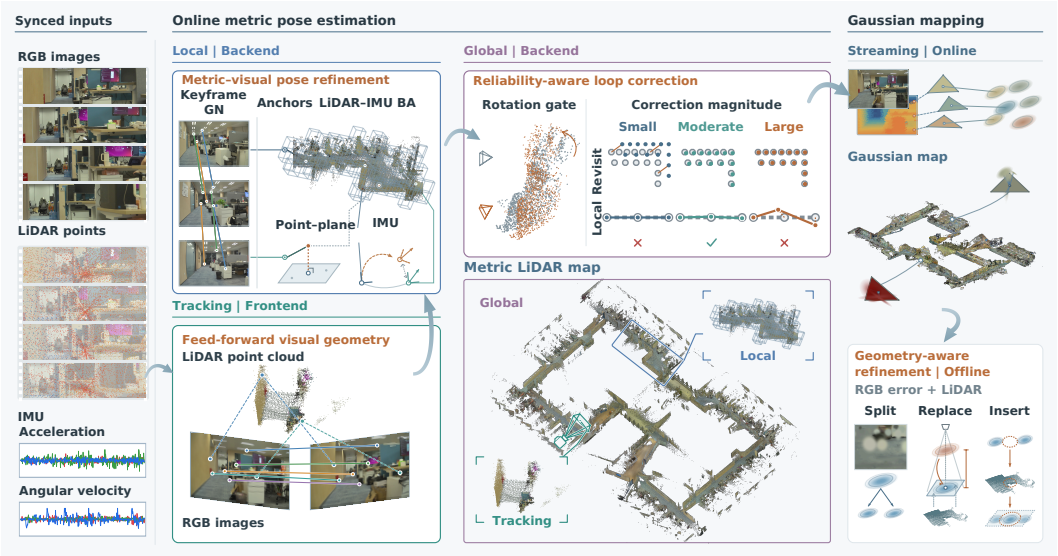

Figure 2: **Pipeline of MIRAGE.** We anchor feed-forward visual geometry to LiDAR–IMU motion for metric pose estimation and streaming Gaussian mapping. Local metric constraints support visual pose refinement, while reliability-aware loop correction controls correction magnitude and selectively applies rotation to balance revisit alignment and local motion consistency. The resulting poses drive a triangle-associated Gaussian map. With poses fixed, offline geometry-aware refinement improves scene structure using image residuals and LiDAR support.

spondence support and cross-modal agreement regulate pose corrections (Section 3.2). The resulting poses drive streaming Gaussian reconstruction. After streaming, we fix camera poses and refine the Gaussian map offline, using image residuals and LiDAR depth to guide primitive splitting, replacement, and insertion (Section 3.3). We also introduce a synchronized multimodal indoor benchmark with MIRAGE-derived 6-DoF reference trajectories for evaluating long-horizon monocular reconstruction (Section 3.4).

3.2 Pose Estimation

We anchor visual pose estimation to LiDAR–IMU motion through a local-to-global pipeline. **Tracking** combines dense visual correspondences with LiDAR-derived metric depth to estimate frame-to-keyframe poses (Section 3.2.1). **Local optimization** refines keyframe poses using visual consistency constraints and metric position anchors (Section 3.2.2). **Loop closure** selectively corrects these anchors, and **global trajectory fusion** combines them with local visual motion to produce per-image camera poses (Section 3.2.3).

3.2.1 Metric-Guided Visual Tracking

For the current frame f and reference keyframe k, we estimate the frame-to-keyframe transformation $\mathbf{T}^{\mathrm{V}}_{kf} = (\mathbf{T}^{\mathrm{V}}_{k})^{-1}\mathbf{T}^{\mathrm{V}}_{f}$, where $\mathbf{T}^{\mathrm{V}}_{f}, \mathbf{T}^{\mathrm{V}}_{k} \in \mathrm{SE}(3)$ are their visual camera-to-world poses. As shown in Figure 2, the feed-forward model (Leroy et al., 2024) predicts dense image correspondences and their confidence scores, while LiDAR projection and depth completion (Wang et al., 2025c) provide dense metric depth. We optimize $\mathbf{T}^{\mathrm{V}}_{kf}$ by minimizing a Huber loss over confidence-weighted pixel reprojection and log-depth residuals using Gauss–Newton, following (Murai et al., 2025). Details are provided in Appendix A.2.1.

3.2.2 Local Optimization with LiDAR–Inertial Support

After tracking and keyframe insertion, we couple visual keyframe optimization with LiDAR–inertial bundle adjustment through metric position anchors. Visual correspondences constrain inter-frame

image and depth consistency, while LiDAR and IMU observations constrain metric motion. We first optimize the visual camera-to-world poses $\mathbf{T}_j^{\mathrm{V}} = (\mathbf{R}_j^{\mathrm{V}}, \mathbf{p}_j^{\mathrm{V}}) \in \mathrm{SE}(3)$ by minimizing

$$E_{\mathrm{V}} = \sum_{(j,k)\in\mathcal{E}_{\mathrm{V}}} \sum_{m\in\mathcal{M}_{jk}} \rho\big(\mathbf{W}_{jk,m}\, \mathbf{r}_{jk,m}^{\mathrm{V}}\big((\mathbf{T}_k^{\mathrm{V}})^{-1}\mathbf{T}_j^{\mathrm{V}}\big)\big)\,. \tag{1}$$

Here, $\mathcal{E}_{\mathrm{V}}$ contains directed keyframe connections, and $\mathcal{M}_{jk}$ indexes valid correspondences. Using per-image metric depth, $\mathbf{r}^{\mathrm{V}}$ combines pixel reprojection and log-depth residuals; $\mathbf{W}$ encodes confidence and residual scaling, and ρ sums componentwise Huber penalties. We fix the first keyframe pose to remove gauge freedom.

For LiDAR–inertial support, we organize multi-scan points into adaptive voxels and fit shared planes using PCA on locally planar clusters (Liu et al., 2026). We separately refine motion within a sliding window by minimizing $E_{\mathrm{LI}} = E_{\mathrm{LiDAR}} + E_{\mathrm{IMU}}$. The LiDAR term penalizes point spread perpendicular to each plane, adjusting scan poses to reduce local point-cloud thickness and align observations across scans. The IMU term penalizes disagreement between estimated motion and integrated gyroscope and accelerometer measurements.

We interpolate the optimized LiDAR–inertial poses to keyframe timestamps, convert them to camera poses using calibrated extrinsics, and rigidly align them to the visual world frame. The resulting poses $\mathbf{T}_j^{\mathrm{M}} = (\mathbf{R}_j^{\mathrm{M}}, \mathbf{p}_j^{\mathrm{M}})$ serve as metric anchors. We then refine visual keyframe positions by adding $E_{\mathrm{anchor}} = \sum_{j\in\mathcal{S}} \|\mathbf{p}_j^{\mathrm{V}} - \mathbf{p}_j^{\mathrm{M}}\|_2^2/\sigma_{\mathrm{BA}}^2$, where $\mathcal{S}$ indexes keyframes with available anchors and σ_{BA} controls the prior strength. Anchor positions remain fixed during refinement. Full pose anchors are retained for subsequent global correction and trajectory fusion (Section 3.2.3). Details are provided in Appendix A.2.2.

3.2.3 Loop Closure and Global Trajectory Fusion

We selectively update the metric camera anchors through **observation-supported loop correction** and combine them with the locally refined visual trajectory through **metric-anchored visual motion fusion** to obtain per-image camera poses.

Observation-supported loop correction. We generate initial loop candidates following (Liu et al., 2026) and rigidly align the resulting pose-graph optimization (PGO) proposals to the current metric trajectory without scale fitting. We assess each proposal from two complementary perspectives: (1) *geometric consistency*, evaluating improvements in revisit alignment and the risk of disrupting local structure; and (2) *visual consistency*, evaluating reprojection agreement with available image correspondences. These cues determine the accepted correction magnitude. Rotation updates are gated separately and require additional geometric benefit relative to their risk; otherwise, the current metric orientations are retained, as shown in Figure 2. Accepted corrections update the metric anchors and propagate to subsequent metric poses. Details are provided in Appendix A.2.3.

Metric-anchored visual motion fusion. We interpolate consecutive metric anchors to obtain a baseline camera trajectory. We then incorporate visual motion between these anchors, using the deviation of the visual trajectory from the interpolation of its poses at the same two anchor timestamps. Let $\mathbf{T}^{\mathrm{M}}(t)$ interpolate the accepted metric anchors, $\mathbf{T}^{\mathrm{V}}(t)$ denote the locally refined visual trajectory, and $\overline{\mathbf{T}}^{\mathrm{V}}(t)$ interpolate the visual poses at the same two anchor timestamps. We obtain the fused camera pose $\mathbf{T}^{\mathrm{F}}(t)$ by combining their rotation and position components:

$$\mathbf{T}^{\mathrm{F}}(t) = \Big(\mathbf{R}^{\mathrm{M}}(t)\overline{\mathbf{R}}^{\mathrm{V}}(t)^{\top}\mathbf{R}^{\mathrm{V}}(t),\ \mathbf{p}^{\mathrm{M}}(t) + \mathbf{p}^{\mathrm{V}}(t) - \overline{\mathbf{p}}^{\mathrm{V}}(t)\Big)\,. \tag{2}$$

The fused trajectory matches the metric anchors exactly while retaining visual motion residuals between them. Details are provided in Appendix A.2.4. We optionally refine the accepted metric trajectory offline using hierarchical LiDAR bundle adjustment (Liu et al., 2023), exporting a separate trajectory as described in Appendix A.2.5.

3.3 Geometry-Aware Gaussian Reconstruction

We reconstruct scenes using LiDAR to constrain geometry and RGB images to supervise appearance. Inspired by PLANING (Jiang et al., 2026), we use a triangle–Gaussian representation that couples explicit surface geometry with Gaussian appearance.

Streaming Gaussian reconstruction. For each incoming keyframe, we initialize triangle geometry from LiDAR depth and normals, and Gaussian appearance from RGB observations. We then jointly optimize geometry and appearance through differentiable rendering: depth and normal consistency constrain geometry, while photometric losses supervise appearance. When camera poses are updated, we transform the associated primitives accordingly to maintain map consistency. Details are provided in Appendix A.3.

Geometry-aware Gaussian refinement. Following prior work (Li et al., 2026b; Meuleman et al., 2025; Zhang et al., 2025), we refine the streamed scene offline to improve global scene consistency and reconstruction accuracy. With camera poses fixed, LiDAR depth constrains geometry, while RGB images refine appearance. During refinement, we periodically diagnose and repair regions with high rendering error using motion-compensated LiDAR depth and multi-view consistency checks. (1) Oversized Gaussians whose depths agree with LiDAR measurements but fail to resolve image detail are **split** into smaller primitives. (2) Gaussians with persistent depth disagreement are **replaced** with primitives reinitialized using LiDAR depth and local normals. (3) At locations with valid LiDAR depth but insufficient Gaussian coverage, we **insert** new primitives using depth for position, local normals for orientation, and RGB observations for color. Alongside the photometric loss, we apply a point-to-plane penalty constructed from LiDAR depth and local normals to newly created or reinitialized Gaussians, penalizing position errors along the normal direction while allowing tangential adjustment. Repair criteria and loss details are provided in Appendix A.3.2.

3.4 A Benchmark for Long-Horizon Indoor Reconstruction

We introduce a benchmark for long-horizon monocular streaming reconstruction. Our platform (Fig. 1, right) integrates a wide-angle RGB camera, two LiDARs, and one LiDAR's built-in IMU. The dual-LiDAR configuration combines broad coverage for robust localization with dense sampling for detailed reconstruction. Custom hardware synchronization and calibrated extrinsics align the sensor streams. We collected sequences through low-speed teleoperation to limit motion blur and removed blurred images through manual inspection.

The dataset comprises 10 sequences across four categories: laboratories and dining areas with numerous objects and reflective surfaces, an office, and long, low-texture corridors that challenge visual tracking. It contains 71,536 RGB images at 2880×1860 resolution, spanning 59.6 minutes and 1.15 km, with trajectories approaching 200 m. We provide synchronized RGB–LiDAR–IMU data, sensor intrinsics and extrinsics, global LiDAR reference point clouds, and MIRAGE-derived 6-DoF reference poses. MIRAGE's trajectory accuracy is validated on Oxford Spires (Tao et al., 2026) and M2DGR (Yin et al., 2022). Platform details and dataset statistics appear in Appendix A.4.

We evaluate RGB-only methods on global trajectory accuracy and downstream reconstruction quality. We report absolute trajectory error (ATE) RMSE after Sim(3) alignment to the reference poses. For reconstruction evaluation, we train standard 3DGS (Kerbl et al., 2023) with each method's predicted poses held fixed. Held-out novel view synthesis (NVS) PSNR serves as a downstream indicator of local multi-view pose consistency.

4 Experiments

4.1 Experimental Setup

Datasets and Metrics. We evaluate MIRAGE on Oxford Spires (Tao et al., 2026), M2DGR (Yin et al., 2022), and INS (Deng et al., 2025a), reporting absolute trajectory error (ATE) RMSE for trajectory accuracy and PSNR, SSIM, and LPIPS (Zhang et al., 2018) for rendering quality. We additionally evaluate representative vision-only methods on our proposed MIRAGE benchmark, reporting ATE RMSE after Sim(3) alignment to the reference trajectories we provide. Details are provided in Appendix A.5.

Baselines. On Oxford Spires, M2DGR, and INS, our comparisons include LiDAR–inertial odometry (EllipseLIO (Border & Chli, 2026)), LiDAR–inertial–visual odometry (Fast-LIVO2 (Zheng et al., 2025)), LiDAR–inertial SLAM without visual input (Fast-LIO-SLAM and Voxel-SLAM (Liu et al., 2026)), and vision-only methods (LingBot-Map (Chen et al., 2026), GLUEMAP (Pan et al., 2026), and COLMAP (Schönberger & Frahm, 2016)). We also compare with Gaussian-LIC2 (Lang

Table 1: Translational ATE RMSE (cm, ↓). Vision-only and metric methods use Sim(3) and SE(3) alignment, respectively. Red/orange mark the best/second-best per sequence, ranked separately for online and offline methods; – denotes failure.

	Online						Offline			
Sequence	EllipseLIO	Fast-LIO-SLAM	Fast-LIVO2	Gaussian-LIC2	LingBot-Map	**Ours**	Voxel-SLAM	COLMAP	GLUEMAP	**Ours+HBA**
Latency	0.10	0.12	0.10	0.71	0.18	0.73	0.145	1.85	10.65	0.76
Oxford Spires										
Observatory 01	5.74	6.28	3.97	6.41×10^{4}	2.60×10^{3}	3.43	2.46	6.66	95.79	2.09
Observatory 02	5.23	26.22	6.83	–	2.52×10^{3}	3.56	2.14	7.69	75.81	2.07
Keble 02	6.02	37.35	3.01×10^{5}	3.08×10^{2}	3.15×10^{3}	3.60	3.48	4.74	3.30×10^{3}	3.66
Keble 03	6.02	33.26	7.92	2.32×10^{8}	2.59×10^{3}	7.13	3.60	4.51	3.09×10^{3}	3.07
Keble 04	19.14	46.38	8.82	1.50×10^{5}	4.01×10^{3}	9.61	8.37	7.03	1.14×10^{2}	6.52
Blenheim 01	23.58	58.42	14.04	1.17×10^{4}	5.27×10^{3}	4.50	3.88	7.95	5.65×10^{3}	3.52
Blenheim 02	11.20	11.30	23.18	1.99×10^{5}	4.61×10^{3}	3.27	2.50	5.50	3.83×10^{3}	2.31
Church 03	2.05	29.44	2.73×10^{6}	–	1.67×10^{3}	1.15	1.35	6.86	3.77×10^{2}	1.50
M2DGR										
room_01	2.87	24.57	2.59	2.97	71.23	1.87	1.72	31.26	23.84	1.67
room_02	2.65	19.27	2.40	2.64	72.29	2.18	1.89	11.56	27.71	1.90
room_03	2.43	24.62	2.91	3.86	68.85	2.15	2.10	15.69	15.55	2.00
INS										
indoor_sq1	10.21	9.96	11.22	15.37	1.85×10^{3}	5.67	4.02	52.99	1.95×10^{3}	2.96
indoor_sq2	4.35	10.19	15.16	–	1.81×10^{3}	5.48	4.38	–	1.93×10^{3}	3.37
indoor_sq3	6.03	6.50	10.52	11.62	1.89×10^{3}	5.20	3.92	52.33	1.88×10^{3}	3.70
indoor_sq4	4.77	8.30	14.20	12.28	1.93×10^{3}	4.44	3.90	129.19	2.56×10^{2}	4.03

et al., 2026), a tightly coupled LiDAR–inertial–visual Gaussian SLAM system. On our MIRAGE benchmark, we evaluate recent strong vision-only methods, including LingBot-Map, GLUEMAP, LoGeR (Zhang et al., 2026), Scal3R (Xie et al., 2026), and ABot-Recon (Han et al., 2026), alongside MASt3R-SLAM (Murai et al., 2025).

Implementation Details. Following the 3DGS evaluation protocol (Kerbl et al., 2023), we hold out every eighth image for rendering evaluation. LiDAR–inertial local bundle adjustment uses a ten-scan sliding window. Gaussian refinement runs for 15,000 steps with camera poses and exposure transforms fixed, performing splitting every 1,000 steps from step 2,000 through 8,000. All experiments are conducted on NVIDIA RTX 4090 GPUs.

4.2 Trajectory Accuracy

MIRAGE completes all 15 sequences and achieves the lowest reported online ATE on 12 while maintaining a moderate latency of 0.73 s/frame (Table 1). Averaged equally across sequences, its ATE is 4.22 cm versus 7.49 cm for EllipseLIO, a 43.7% reduction. On large-scale Oxford Spires traversals and low-texture INS scenes, LingBot-Map exhibits substantial trajectory errors, while Fast-LIVO2 and Gaussian-LIC2 show severe errors or failures in our reproductions. MIRAGE maintains ATE below 10 cm on every sequence. These results are consistent with its reliability-aware design: metric anchors constrain drift, while correspondence support and cross-modal agreement regulate visual corrections to exploit useful evidence and limit unreliable updates. With offline HBA, MIRAGE achieves a mean ATE of 2.96 cm versus 3.31 cm for Voxel-SLAM, with lower errors on 11 of 15 sequences.

4.3 Rendering Quality

We compare MIRAGE with Gaussian-LIC2 and offline 3DGS trained from scratch using MIRAGE trajectories (Table 2). We evaluate 12 sequences, excluding three for which Gaussian-LIC2 did not complete reconstruction. MIRAGE achieves higher PSNR than Gaussian-LIC2 on 10 of 12 sequences, with a mean gain of 3.27 dB. Using identical camera trajectories, our refined streaming maps also exceed offline 3DGS by 1.15 dB in mean PSNR. These results are consistent with the complementary benefits of reliable multi-view alignment and Gaussian refinement that combines

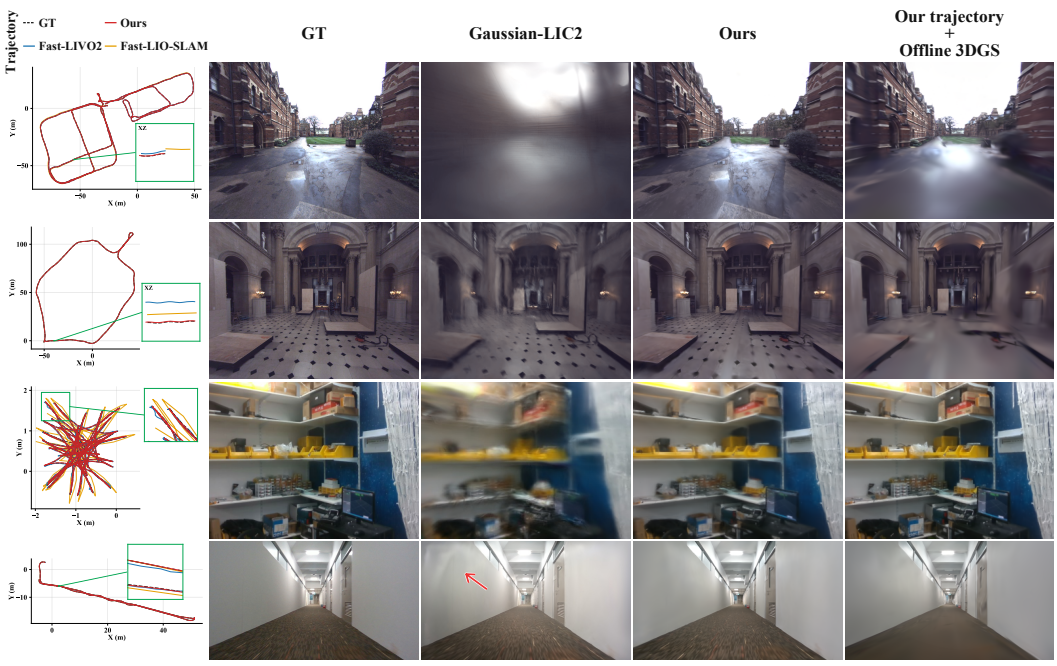

Figure 3: **Qualitative trajectory and rendering comparisons.** Left: SE(3)-aligned XY trajectories; green boxes show XZ close-ups with equal axis scales. Right: rendering comparisons.

Table 2: Held-out rendering quality. Offline 3DGS (Kerbl et al., 2023) uses 30k iterations with MIRAGE's online poses fixed. Cells list PSNR (dB, ↑), SSIM (↑), and LPIPS (↓) from top to bottom. Red and orange indicate the best and second-best values per sequence and metric.

Method	*Oxford Spires*						*M2DGR*			*INS*		
	Observatory 01	Keble 02	Keble 03	Keble 04	Blenheim 01	Blenheim 02	room_01	room_02	room_03	indoor_sq1	indoor_sq3	indoor_sq4
Gaussian-LIC2	17.91	24.26	17.49	18.72	21.76	21.28	25.15	26.67	24.60	30.09	29.24	30.45
	0.678	0.810	0.678	0.680	0.738	0.744	0.809	0.829	0.770	0.893	0.885	0.896
	0.570	0.421	0.554	0.548	0.530	0.535	0.258	0.251	0.341	0.299	0.308	0.301
Ours	26.26	27.53	26.80	24.33	26.40	27.82	25.44	26.76	26.60	30.03	29.58	29.34
	0.827	0.836	0.821	0.741	0.796	0.837	0.824	0.831	0.843	0.907	0.901	0.900
	0.308	0.325	0.344	0.390	0.396	0.357	0.255	0.252	0.249	0.294	0.306	0.307
Our trajectory + Offline 3DGS	26.04	26.87	25.75	23.19	24.82	26.65	26.69	27.02	26.42	26.36	27.71	25.60
	0.801	0.823	0.804	0.711	0.749	0.787	0.865	0.847	0.850	0.831	0.855	0.832
	0.389	0.366	0.400	0.476	0.507	0.470	0.212	0.231	0.248	0.451	0.414	0.453

LiDAR-guided structural repair with RGB appearance supervision to recover scene detail. Figure 3 presents representative trajectory and rendering comparisons across the three datasets.

4.4 Ablation Studies

Tracking, Local Optimization, and Global Correction. Table 3(a) shows that local optimization substantially reduces mean ATE across datasets, while global correction provides further gains with occasional small regressions. Oxford Spires shows the largest overall improvement, with mean ATE decreasing from 82.36 cm after tracking to 4.53 cm after global correction. Offline HBA results are included as a separate reference.

Visual Evidence and Gaussian Refinement. Table 3(b) shows that removing the visual pose and geometry branch substantially degrades rendering, highlighting the joint contribution of visual tracking and dense geometric priors. Both refinement variants improve upon the full streaming model.

Table 3: Pose and rendering ablations. (a) Scene-averaged ATE (cm, ↓) over all evaluated sequences. (b) PSNR (dB, ↑) / SSIM (↑) / LPIPS (↓). †: camera poses are obtained solely by interpolating LiDAR–inertial poses and applying calibrated extrinsics; RGB supervision for Gaussian reconstruction is retained.

(a) Pose estimation				**(b) Rendering on M2DGR**			
Stage	Oxford Spires	M2DGR	INS	Configuration	room_01	room_02	room_03
Tracking	82.363	21.868	43.819	w/o visual†	13.90 / 0.504 / 0.561	13.04 / 0.512 / 0.559	12.89 / 0.489 / 0.577
Local	7.784	1.909	7.055	Streaming	23.28 / 0.746 / 0.339	24.41 / 0.762 / 0.344	24.74 / 0.784 / 0.319
Global	4.529	2.067	5.195	+ Photo. refine	24.96 / 0.811 / 0.272	26.29 / 0.816 / 0.275	26.22 / 0.832 / 0.265
Offline (+HBA)	3.093	1.858	3.513	+ Geo. refine	25.44 / 0.824 / 0.255	26.76 / 0.831 / 0.252	26.60 / 0.843 / 0.249

Table 4: MIRAGE benchmark. ATE: Sim(3)-aligned translational RMSE against MIRAGE references. Offline 3DGS: held-out PSNR with fixed poses; Ours (reference) uses reference poses. Stream PSNR: MIRAGE after refinement. Red/orange indicate the best/second-best per sequence within the ATE and offline 3DGS groups; – denotes trajectory failure.

		ATE (m, ↓)						Offline 3DGS PSNR (dB, ↑)							Stream PSNR
Sequence	Trajectory length (m)	LingBot-Map	GlueMap	LoGeR	Scal3R	ABot-Recon	MASt3R-SLAM	LingBot-Map	GlueMap	LoGeR	Scal3R	ABot-Recon	MASt3R-SLAM	Ours (reference)	Ours
Laboratory 1	54.57	5.979	0.205	0.927	1.901	0.368	0.491	17.40	26.59	20.74	22.16	23.28	17.54	26.10	27.47
Laboratory 2	61.02	6.034	0.225	0.802	1.102	0.675	0.866	17.20	26.41	19.94	22.11	22.53	14.59	25.02	28.47
Laboratory 3	168.46	18.954	15.851	7.364	1.662	0.930	0.729	16.55	22.60	18.96	21.72	22.06	17.22	23.56	26.74
Dining Area 1	112.69	9.694	15.723	10.616	14.772	0.903	2.168	15.23	22.16	18.76	19.09	21.33	16.16	21.68	26.15
Dining Area 2	64.65	8.616	0.206	1.511	0.350	0.427	0.412	16.25	20.45	19.45	20.50	20.87	16.41	19.84	24.06
Dining Area 3	181.52	11.001	0.746	6.447	10.129	1.656	3.836	15.65	22.35	18.34	20.87	21.65	15.34	22.91	26.42
Dining Area 4	65.37	8.600	0.149	0.376	0.602	0.361	0.790	18.54	25.72	21.43	24.33	24.10	16.56	24.69	26.72
Corridor 1	165.39	18.042	13.560	13.153	6.802	1.873	–	19.07	23.47	21.01	23.76	23.86	–	25.41	29.12
Corridor 2	85.57	9.027	11.387	7.694	9.206	1.026	–	19.99	22.29	23.02	25.83	26.24	–	27.87	30.89
Office 1	195.15	19.164	23.101	16.040	21.136	7.600	–	15.18	15.92	15.68	16.25	17.80	–	18.57	24.35

Under the same 15k-step budget with poses and exposure fixed, geometry-aware refinement exceeds photometric refinement by 0.44 dB in mean PSNR, with higher SSIM and lower LPIPS on all three sequences.

4.5 Vision-Only Evaluation on the MIRAGE Benchmark

Table 4 evaluates RGB-only pose estimates through global trajectory accuracy and downstream fixed-pose 3DGS rendering. ABot-Recon achieves the lowest mean Sim(3)-aligned ATE among methods completing all ten sequences (1.58 m), while GLUEMAP achieves the highest mean downstream PSNR among visual baselines (22.80 dB). Their differing rankings suggest that global trajectory agreement alone does not determine reconstruction quality: local multi-view pose consistency also matters. Long indoor traversals, repetitive structures, and low-texture corridors challenge drift control and correspondence estimation, exposing complementary strengths and limitations of current visual methods.

5 Conclusion

We presented MIRAGE, which anchors feed-forward visual priors to LiDAR–IMU motion for metric pose estimation and Gaussian reconstruction. Visual and geometric evidence guide trajectory corrections, while LiDAR depth and image residuals drive Gaussian repair with camera poses fixed. Experiments on Oxford Spires, M2DGR, and INS demonstrate accurate trajectories and high-fidelity rendering. Our synchronized multimodal benchmark comprises 10 sequences across four indoor scene categories, with 71,536 RGB images spanning 59.6 minutes and 1.15 km. With MIRAGE-derived 6-DoF reference trajectories, sensor calibration, and global LiDAR maps, the benchmark provides the community with a common testbed for evaluating trajectory accuracy and downstream novel-view synthesis in monocular streaming reconstruction, supporting further advances in long-horizon indoor reconstruction.

6 AI USE STATEMENT

In this work, we used generative AI tools solely to assist with language polishing of the manuscript, including improving grammar, clarity, and readability. We did not use generative AI tools to generate research ideas, develop methods, analyze data, or produce experimental results. We have reviewed all AI-assisted edits to ensure that they preserve the intended meaning and technical accuracy of the manuscript and do not introduce unsupported claims. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

REFERENCES

Rowan Border and Margarita Chli. EllipseLIO: Adaptive LiDAR inertial odometry with an ellipsoid representation. *arXiv preprint arXiv:2605.21150*, 2026.

Kenny Chen, Ryan Nemiroff, and Brett T Lopez. Direct lidar-inertial odometry: Lightweight lio with continuous-time motion correction. *arXiv preprint arXiv:2203.03749*, 2022.

Lin-Zhuo Chen, Jian Gao, Yihang Chen, Ka Leong Cheng, Yipengjing Sun, Liangxiao Hu, Nan Xue, Xing Zhu, Yujun Shen, Yao Yao, and Yinghao Xu. Geometric context transformer for streaming 3D reconstruction. In *European Conference on Computer Vision (ECCV)*, 2026.

Zequn Chen, Jiezhi Yang, and Heng Yang. PreF3R: Pose-free feed-forward 3D gaussian splatting from variable-length image sequence. *arXiv preprint arXiv:2411.16877*, 2024.

Jiadi Cui, Junming Cao, Fuqiang Zhao, Zhipeng He, Yifan Chen, Yuhui Zhong, Lan Xu, Yujiao Shi, Yingliang Zhang, and Jingyi Yu. Letsgo: Large-scale garage modeling and rendering via lidar-assisted gaussian primitives. *ACM Transactions on Graphics (TOG)*, 43(6):1–18, 2024.

Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Niessner. ScanNet: Richly-annotated 3D reconstructions of indoor scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5828–5839, 2017.

Junyuan Deng, Heng Li, Kejie Qiu, Lingteng Qiu, Rui Peng, Weichao Shen, Weihao Yuan, Siyu Zhu, Zilong Dong, and Ping Tan. Glob3R: Global structure-from-motion with 3D foundation models. *arXiv preprint arXiv:2607.09225*, 2026.

Tianchen Deng, Guole Shen, Chen Xun, Shenghai Yuan, Tongxin Jin, Hongming Shen, Yanbo Wang, Jingchuan Wang, Hesheng Wang, Danwei Wang, and Weidong Chen. MNE-SLAM: Multi-agent neural SLAM for mobile robots. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1485–1494, 2025a.

Tianchen Deng, Wenhua Wu, Junjie He, Yue Pan, Shenghai Yuan, Danwei Wang, and Hesheng Wang. Vpgs-slam: Voxel-based progressive 3d gaussian slam in large-scale scenes. *arXiv preprint arXiv:2505.18992*, 2025b.

Jiarong Han, Jincheng Xiong, Yuzhou Liu, Linzhe Shi, Changjie Wu, Ning Guo, Mu Xu, Hang Zhang, and Ming Qian. Revisiting local context for long-horizon streaming 3D reconstruction. *arXiv preprint arXiv:2608.27529*, 2026.

Sheng Hong, Junjie He, Xinhu Zheng, and Chunran Zheng. LIV-GaussMap: LiDAR-inertial-visual fusion for real-time 3D radiance field map rendering. *IEEE Robotics and Automation Letters*, 9 (11):9765–9772, 2024. doi: 10.1109/LRA.2024.3400149.

Sheng Hong, Chunran Zheng, Yishu Shen, Changze Li, Fu Zhang, Tong Qin, and Shaojie Shen. GS-LIVO: Real-time LiDAR, inertial, and visual multisensor fused odometry with gaussian mapping. *IEEE Transactions on Robotics*, 41:4253–4268, 2025. doi: 10.1109/TRO.2025.3582809.

Huajian Huang, Longwei Li, Hui Cheng, and Sai-Kit Yeung. Photo-SLAM: Real-time simultaneous localization and photorealistic mapping for monocular, stereo, and RGB-D cameras. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21584–21593, 2024.

Changjian Jiang, Ruilan Gao, Kele Shao, Yue Wang, Rong Xiong, and Yu Zhang. Li-gs: Gaussian splatting with lidar incorporated for accurate large-scale reconstruction. *IEEE Robotics and Automation Letters*, 10(2):1864–1871, 2024.

Changjian Jiang, Lijie Wang, Zeyu Wan, Ruilan Gao, Yue Wang, Rong Xiong, and Yu Zhang. Rised: Accurate and efficient rgb-colorized mapping using image selection and point cloud densification. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3277–3283. IEEE, 2025.

Changjian Jiang, Kerui Ren, Xudong Li, Kaiwen Song, Guanghao Li, Linning Xu, Tao Lu, Junting Dong, Yu Zhang, Bo Dai, et al. Planing: A loosely coupled triangle-gaussian framework for streaming 3d reconstruction. *arXiv preprint arXiv:2601.22046*, 2026.

Jianhao Jiao, Hexiang Wei, Tianshuai Hu, Xiangcheng Hu, Yilong Zhu, Zhijian He, Jin Wu, Jingwen Yu, Xupeng Xie, Huaiyang Huang, Ruoyu Geng, Lujia Wang, and Ming Liu. FusionPortable: A multi-sensor campus-scene dataset for evaluation of localization and mapping accuracy on diverse platforms. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3851–3856, 2022. doi: 10.1109/IROS47612.2022.9982119.

Nikhil Keetha, Jay Karhade, Krishna Murthy Jatavallabhula, Gengshan Yang, Sebastian Scherer, Deva Ramanan, and Jonathon Luiten. SplaTAM: Splat, track & map 3D gaussians for dense RGB-D SLAM. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21357–21366, 2024.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkuehler, and George Drettakis. 3D gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4):139:1–139:14, 2023. doi: 10.1145/3592433.

Giseop Kim, Yeong Sang Park, Younghun Cho, Jinyong Jeong, and Ayoung Kim. MulRan: Multimodal range dataset for urban place recognition. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6246–6253, 2020. doi: 10.1109/ICRA40945.2020.9197298.

Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics*, 36(4), 2017. doi: 10.1145/3072959.3073599.

Xiaolei Lang, Laijian Li, Chenming Wu, Chen Zhao, Lina Liu, Yong Liu, Jiajun Lv, and Xingxing Zuo. Gaussian-LIC: Real-time photo-realistic SLAM with gaussian splatting and LiDAR-inertial-camera fusion. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 8500–8507, 2025. doi: 10.1109/ICRA55743.2025.11128712.

Xiaolei Lang, Jiajun Lv, Kai Tang, Laijian Li, Jianxin Huang, Lina Liu, Yong Liu, and Xingxing Zuo. Gaussian-LIC2: LiDAR-inertial-camera gaussian splatting SLAM. *The International Journal of Robotics Research*, 2026. Accepted; preprint arXiv:2507.04004.

Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3d with mast3r. In *European conference on computer vision*, pp. 71–91. Springer, 2024.

Dongen Li, Yi Liu, Junqi Liu, Zewen Sun, Zefan Huang, Shuo Sun, Jiahui Liu, Chengran Yuan, Hongliang Guo, Francis E. H. Tay, and Marcelo H. Ang, Jr. RMGS-SLAM: Real-time multi-sensor gaussian splatting SLAM. *arXiv preprint arXiv:2604.12942*, 2026a.

Guanghao Li, Kerui Ren, Linning Xu, Zhewen Zheng, Changjian Jiang, Xin Gao, Bo Dai, Jian Pu, Mulin Yu, and Jiangmiao Pang. Artdeco: Towards efficient and high-fidelity on-the-fly 3d reconstruction with structured scene representation. In *The Fourteenth International Conference on Learning Representations*, 2026b.

Yang Li, Jinglu Wang, Lei Chu, Xiao Li, Shiu-Hong Kao, Ying-Cong Chen, and Yan Lu. Streamgs: Online generalizable gaussian splatting reconstruction for unposed image streams. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 25841–25850, 2025.

Chin-Yang Lin, Cheng Sun, Fu-En Yang, Min-Hung Chen, Yen-Yu Lin, and Yu-Lun Liu. LongSplat: Robust unposed 3D gaussian splatting for casual long videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 27412–27422, 2025a.

Haotong Lin, Sili Chen, Junhao Liew, Donny Y Chen, Zhenyu Li, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth anything 3: Recovering the visual space from any views. *arXiv preprint arXiv:2511.10647*, 2025b.

Jiarong Lin and Fu Zhang. R3LIVE: A robust, real-time, RGB-colored, LiDAR-inertial-visual tightly-coupled state estimation and mapping package. In *2022 International Conference on Robotics and Automation (ICRA)*, 2022.

Xiyuan Liu, Zheng Liu, Fanze Kong, and Fu Zhang. Large-scale LiDAR consistent mapping using hierarchical LiDAR bundle adjustment. *IEEE Robotics and Automation Letters*, 8(3):1523–1530, 2023. doi: 10.1109/LRA.2023.3238902.

Zheng Liu, Haotian Li, Chongjian Yuan, Xiyuan Liu, Jiarong Lin, Rundong Li, Chunran Zheng, Bingyang Zhou, Wenyi Liu, and Fu Zhang. Voxel-SLAM: A complete, accurate, and versatile light detection and ranging-inertial simultaneous localization and mapping system. *Advanced Intelligent Systems*, 8(4):e202501081, 2026. doi: 10.1002/aisy.202501081.

Dominic Maggio and Luca Carlone. Vggt-slam 2.0: Real-time dense feed-forward scene reconstruction. *arXiv preprint arXiv:2601.19887*, 2026.

Dominic Maggio, Hyungtae Lim, and Luca Carlone. Vggt-slam: Dense rgb slam optimized on the sl (4) manifold. *Advances in Neural Information Processing Systems*, 38:129839–129867, 2026.

Hidenobu Matsuki, Riku Murai, Paul H. J. Kelly, and Andrew J. Davison. Gaussian splatting SLAM. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18039–18048, 2024.

Andreas Meuleman, Ishaan Shah, Alexandre Lanvin, Bernhard Kerbl, and George Drettakis. On-the-fly reconstruction for large-scale novel view synthesis from unposed images. *ACM Transactions on Graphics*, 44(4), 2025.

Andreas Meuleman, Linus Franke, Boris Zhestiankin, Camille Montemagni, and George Drettakis. Immediate 3d gaussian splat reconstruction of unordered input with global consistency. In *SIGGRAPH Conference Papers '26*, pp. 188:1–188:12, 2026. doi: 10.1145/3799902.3811167.

Riku Murai, Eric Dexheimer, and Andrew J. Davison. MASt3R-SLAM: Real-time dense SLAM with 3D reconstruction priors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16695–16705, 2025.

Linfei Pan, Johannes Lutz Schönberger, and Marc Pollefeys. Global structure-from-motion meets feedforward reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

Kerui Ren, Guanghao Li, Changjian Jiang, Yingxiang Xu, Tao Lu, Linning Xu, Junting Dong, Jiangmiao Pang, Mulin Yu, and Bo Dai. M3: Dense matching meets multi-view foundation models for monocular gaussian splatting slam. *arXiv preprint arXiv:2603.16844*, 2026.

Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

Tixiao Shan, Brendan Englot, Drew Meyers, Wei Wang, Carlo Ratti, and Daniela Rus. LIO-SAM: Tightly-coupled lidar inertial odometry via smoothing and mapping. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5135–5142, 2020.

Tixiao Shan, Brendan Englot, Carlo Ratti, and Daniela Rus. LVI-SAM: Tightly-coupled lidar-visual-inertial odometry via smoothing and mapping. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5692–5698, 2021.

Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. A benchmark for the evaluation of RGB-D SLAM systems. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 573–580, 2012. doi: 10.1109/IROS.2012.6385773.

Yifu Tao, Miguel Ángel Muñoz-Bañón, Lintong Zhang, Jiahao Wang, Lanke Frank Tarimo Fu, and Maurice Fallon. The Oxford Spires Dataset: Benchmarking large-scale LiDAR-visual localisation, reconstruction and radiance field methods. *The International Journal of Robotics Research*, 45(6):839–857, 2026. doi: 10.1177/02783649251369905.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025a.

Lijie Wang, Lianjie Guo, Ziyi Xu, Qianhao Wang, Fei Gao, and Xieyuanli Chen. LiDAR-VGGT: Cross-modal coarse-to-fine fusion for globally consistent and metric-scale dense mapping. *IEEE Robotics and Automation Letters*, 11(4):4721–4728, 2026. doi: 10.1109/LRA.2026.3666387.

Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. DUSt3R: Geometric 3D vision made easy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Yifan Wang, Jianjun Zhou, Haoyi Zhu, Wenzheng Chang, Yang Zhou, Zizun Li, Junyi Chen, Jiangmiao Pang, Chunhua Shen, and Tong He. pi3: Permutation-equivariant visual geometry learning. *arXiv preprint arXiv:2507.13347*, 2025b.

Zehan Wang, Siyu Chen, Lihe Yang, Jialei Wang, Ziang Zhang, Hengshuang Zhao, and Zhou Zhao. Depth anything with any prior. *arXiv preprint arXiv:2505.10565*, 2025c.

Hexiang Wei, Jianhao Jiao, Xiangcheng Hu, Jingwen Yu, Xupeng Xie, Jin Wu, Yilong Zhu, Yuxuan Liu, Lujia Wang, and Ming Liu. FusionPortableV2: A unified multi-sensor dataset for generalized SLAM across diverse platforms and scalable environments. *The International Journal of Robotics Research*, 44(7):1093–1116, 2025. doi: 10.1177/02783649241303525.

Songyin Wu, Zhaoyang Lv, Yufeng Zhu, Duncan Frost, Zhengqin Li, Ling-Qi Yan, Carl Ren, Richard Newcombe, and Zhao Dong. Monocular online reconstruction with enhanced detail preservation. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers*, pp. 1–11, 2025.

Tao Xie, Peishan Yang, Yudong Jin, Yingfeng Cai, Wei Yin, Weiqiang Ren, Qian Zhang, Wei Hua, Sida Peng, Xiaoyang Guo, and Xiaowei Zhou. Scal3R: Scalable test-time training for large-scale 3D reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

Yusen Xie, Zhenmin Huang, Jin Wu, and Jun Ma. GS-LIVM: Real-time photo-realistic LiDAR-inertial-visual mapping with gaussian splatting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 26869–26878, 2025.

Wei Xu, Yixi Cai, Dongjiao He, Jiarong Lin, and Fu Zhang. FAST-LIO2: Fast direct LiDAR-inertial odometry. *IEEE Transactions on Robotics*, 38(4):2053–2073, 2022. doi: 10.1109/TRO.2022.3141876.

Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. ScanNet++: A high-fidelity dataset of 3D indoor scenes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 12–22, 2023.

Jie Yin, Ang Li, Tao Li, Wenxian Yu, and Danping Zou. M2DGR: A multi-sensor and multi-scenario SLAM dataset for ground robots. *IEEE Robotics and Automation Letters*, 7(2):2266–2273, 2022. doi: 10.1109/LRA.2021.3138527.

Junyi Zhang, Charles Herrmann, Junhwa Hur, Chen Sun, Ming-Hsuan Yang, Forrester Cole, Trevor Darrell, and Deqing Sun. LoGeR: Long-context geometric reconstruction with hybrid memory. *arXiv preprint arXiv:2603.03269*, 2026.

Lintong Zhang, Michael Helmberger, Lanke Frank Tarimo Fu, David Wisth, Marco Camurri, Davide Scaramuzza, and Maurice Fallon. Hilti-Oxford dataset: A millimeter-accurate benchmark for simultaneous localization and mapping. *IEEE Robotics and Automation Letters*, 8(1):408–415, 2023. doi: 10.1109/LRA.2022.3226077.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

Wei Zhang, Qing Cheng, David Skuddis, Niclas Zeller, Daniel Cremers, and Norbert Haala. Hi-slam2: Geometry-aware gaussian slam for fast monocular scene reconstruction. *IEEE Transactions on Robotics*, 41:6478–6493, 2025.

Shibo Zhao, Yuanjun Gao, Tianhao Wu, Damanpreet Singh, Rushan Jiang, Haoxiang Sun, Mansi Sarawata, Yuheng Qiu, Warren Whittaker, Ian Higgins, Yi Du, Shaoshu Su, Can Xu, John Keller, Jay Karhade, Lucas Nogueira, Sourojit Saha, Ji Zhang, Wenshan Wang, Chen Wang, and Sebastian Scherer. SubT-MRS dataset: Pushing SLAM towards all-weather environments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22647–22657, 2024.

Shibo Zhao, Sifan Zhou, Yuchen Zhang, Ji Zhang, Chen Wang, Wenshan Wang, and Sebastian Scherer. Resilient odometry via hierarchical adaptation. *Science Robotics*, 10(109):eadv1818, 2025.

Chunran Zheng and Fu Zhang. Fast-calib: Lidar-camera extrinsic calibration in one second. *IEEE Robotics and Automation Practice*, 2026.

Chunran Zheng, Qingyan Zhu, Wei Xu, Xiyuan Liu, Qizhi Guo, and Fu Zhang. Fast-livo: Fast and tightly-coupled sparse-direct lidar-inertial-visual odometry. In *2022 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pp. 4003–4009. IEEE, 2022.

Chunran Zheng, Wei Xu, Zuhao Zou, Tong Hua, Chongjian Yuan, Dongjiao He, Bingyang Zhou, Zheng Liu, Jiarong Lin, Fangcheng Zhu, Yunfan Ren, Rong Wang, Fanle Meng, and Fu Zhang. FAST-LIVO2: Fast, direct LiDAR-inertial-visual odometry. *IEEE Transactions on Robotics*, 41: 326–346, 2025. doi: 10.1109/TRO.2024.3502198.

A Appendix

This appendix is organized as follows:

- Section A.1 defines notation and coordinate conventions.
- Section A.2 details visual tracking, local optimization, loop correction, trajectory fusion, and optional offline pose refinement.
- Section A.3 details streaming Gaussian reconstruction and geometry-aware refinement.
- Section A.4 describes the acquisition platform, calibration, synchronization, collection protocol, and dataset statistics.
- Section A.5 specifies implementation settings and evaluation protocols.
- Section A.6 presents additional experimental results.
- Section A.7 discusses limitations and future directions.

A.1 Notation and Coordinate Conventions

We use bold upright symbols for vectors and matrices. A rigid transformation $\mathbf{T}_{AB} = (\mathbf{R}_{AB}, \mathbf{p}_{AB}) \in \mathrm{SE}(3)$ maps coordinates from frame B to frame A as $\mathbf{x}_A = \mathbf{R}_{AB}\mathbf{x}_B + \mathbf{p}_{AB}$. Camera poses follow the camera-to-world convention, so their translations represent camera centers in world coordinates. Table 5 summarizes the notation used throughout.

Temporal interpolation uses acquisition timestamps τ_i; estimates are consumed only after their availability times a_i. Depth $D_f(\mathbf{u})$ denotes the camera-frame z coordinate of the corresponding 3D point,

Table 5: Notation and coordinate conventions.

Symbol	Definition
$\mathrm{W}, \mathrm{C}, \mathrm{L}, \mathrm{B}$	World, camera, LiDAR, and IMU coordinate frames.
$\mathbf{T}^{\mathrm{V}}, \mathbf{T}^{\mathrm{M}}, \mathbf{T}^{\mathrm{F}}$	Visual camera poses, metric camera-pose anchors, and fused camera poses.
$\mathbf{R} \in \mathrm{SO}(3),\ \mathbf{p} \in \mathbb{R}^3$	Rotation and translation components of a pose.
$f;\ j, k$	Current image index; visual keyframe indices.
$i;\ m$	LiDAR–IMU temporal index; image correspondence index.
$\tau_i,\ a_i$	Sensor acquisition time and corresponding pose-estimate availability time, with $a_i \geq \tau_i$.
$\mathbf{K}$	Calibrated camera intrinsic matrix.
$D_f(\mathbf{u})$	Depth at pixel $\mathbf{u}$ in image f.

rather than its Euclidean distance to the camera center. Additional symbols are defined when first introduced.

A.2 Pose Estimation Details

A.2.1 Metric-Guided Visual Tracking

We use EllipseLIO (Border & Chli, 2026) to estimate the LiDAR–IMU trajectory for motion compensation. Using this trajectory, we align points from five recent LiDAR scans to the camera frame at the image timestamp and project them to obtain sparse metric depth. Depth completion (Wang et al., 2025c) converts the sparse projections into dense depth; invalid pixels are filled from visual depth predictions scaled using valid metric overlap. The feed-forward model (Leroy et al., 2024) supplies image correspondences $(\mathbf{u}_m, \mathbf{v}_m)$ and their confidence scores. The valid set $\mathcal{M}_{fk}$ retains matches passing geometric, confidence, positive-depth, and projection-validity checks.

To optimize the frame-to-keyframe pose $\mathbf{T}^{\mathrm{V}}_{kf}$ using these correspondences and metric depths, we initialize translation from LiDAR–IMU motion expressed in camera coordinates and retain the visual rotation seed. We then jointly refine rotation and translation with the reference keyframe fixed:

$$
\begin{aligned}
\mathbf{T}^{\mathrm{V}}_{kf} &= \arg\min_{\mathbf{T}\in\mathrm{SE}(3)} \sum_{m\in\mathcal{M}_{fk}} \rho\big(\mathbf{W}_{fk,m}\, \mathbf{r}^{\mathrm{V}}_{fk,m}(\mathbf{T})\big)\,, \\
\mathbf{r}^{\mathrm{V}}_{fk,m}(\mathbf{T}) &= \begin{bmatrix} \pi_{\mathbf{K}}(\mathbf{T}\mathbf{X}_f(\mathbf{u}_m)) - \mathbf{v}_m \\ \log \dfrac{[\mathbf{T}\mathbf{X}_f(\mathbf{u}_m)]_z}{D_k(\mathbf{v}_m)} \end{bmatrix}, \\
\mathbf{W}_{fk,m} &= \sqrt{q_{fk,m}}\ \mathrm{diag}\Big(\sigma_{\mathrm{pix}}^{-1}, \sigma_{\mathrm{pix}}^{-1}, \sigma_{\log d}^{-1}\Big)\,.
\end{aligned}
\tag{3}
$$

Here, $\mathbf{X}_f(\mathbf{u}_m) = D_f(\mathbf{u}_m)\mathbf{K}^{-1}\widetilde{\mathbf{u}}_m$ is the back-projected 3D point, with $\widetilde{\mathbf{u}}_m$ denoting homogeneous pixel coordinates. The operator $\pi_{\mathbf{K}}$ performs perspective projection, $[\cdot]_z$ extracts axial depth, and $q_{fk,m}$ is the geometric mean of the matched confidence scores. The scales σ_{pix} and $\sigma_{\log d}$ normalize pixel reprojection and log-depth residuals, respectively, while ρ sums componentwise Huber penalties. Following (Murai et al., 2025), we use Gauss–Newton with Huber reweighting and left updates $\mathbf{T} \leftarrow \exp(\mathbf{d}^{\wedge})\mathbf{T}$, where $\mathbf{d} \in \mathbb{R}^6$ is the pose increment and $(\cdot)^{\wedge}$ maps it to $\mathfrak{se}(3)$. Depths and correspondences remain fixed during each pose solve.

A.2.2 Local Optimization with LiDAR–Inertial Support

Visual keyframe optimization. We optimize visual keyframes using the bidirectional pixel reprojection and log-depth factors in Equation 1, keeping metric depths fixed and fixing the first keyframe pose to remove gauge freedom.

LiDAR–inertial bundle adjustment. Following (Liu et al., 2026), we separately optimize a sliding window of N LiDAR–inertial poses, velocities, and biases:

$$\begin{aligned}\mathcal{X}_{\mathcal{W}} &= \left\{\left(\mathbf{T}_i^{\mathrm{LI}}, \mathbf{v}_i, \mathbf{b}_i^g, \mathbf{b}_i^a\right)\right\}_{i=1}^{N}, \\ \mathcal{X}_{\mathcal{W}}^{\star} &= \arg\min_{\mathcal{X}_{\mathcal{W}}} E_{\mathrm{LiDAR}}(\mathcal{X}_{\mathcal{W}}) + E_{\mathrm{IMU}}(\mathcal{X}_{\mathcal{W}}).\end{aligned} \tag{4}$$

Here, $\mathbf{T}_i^{\mathrm{LI}} = (\mathbf{R}_i, \mathbf{p}_i)$ maps the IMU frame to the LiDAR–inertial world frame; $\mathbf{v}_i$ is the world-frame velocity, and $\mathbf{b}_i^g, \mathbf{b}_i^a$ are gyroscope and accelerometer biases. We transform scan points using these poses and calibrated LiDAR–IMU extrinsics, then group them into adaptive voxels. For each planar voxel $\ell \in \mathcal{V}$, let $\mathcal{P}_\ell$ contain its transformed scan points and fixed historical map points, with $n_\ell = |\mathcal{P}_\ell|$. Its mean, covariance, and contribution to the LiDAR objective are

$$\begin{aligned}\bar{\mathbf{x}}_\ell &= \frac{1}{n_\ell} \sum_{\mathbf{x} \in \mathcal{P}_\ell} \mathbf{x}, \\ \mathbf{C}_\ell(\mathcal{X}_{\mathcal{W}}) &= \frac{1}{n_\ell} \sum_{\mathbf{x} \in \mathcal{P}_\ell} (\mathbf{x} - \bar{\mathbf{x}}_\ell)(\mathbf{x} - \bar{\mathbf{x}}_\ell)^\top, \\ E_{\mathrm{LiDAR}}(\mathcal{X}_{\mathcal{W}}) &= \sum_{\ell \in \mathcal{V}} \lambda_{\min}(\mathbf{C}_\ell(\mathcal{X}_{\mathcal{W}}))\,.\end{aligned} \tag{5}$$

PCA defines a plane through $\bar{\mathbf{x}}_\ell$ whose normal is the eigenvector associated with $\lambda_{\min}$. This eigenvalue equals the mean squared point-to-plane distance. Minimizing it reduces point-cloud thickness and aligns scans to shared planes; the plane parameters are determined analytically from the current poses.

The IMU term constrains consecutive poses and velocities using preintegrated measurements, while accounting for bias changes:

$$\begin{aligned}\mathbf{r}_{i,i+1}^I &= \begin{bmatrix} \mathrm{Log}\left(\Delta\mathbf{R}_i^\top \mathbf{R}_i^\top \mathbf{R}_{i+1}\right) \\ \mathbf{R}_i^\top \left(\mathbf{p}_{i+1} - \mathbf{p}_i - \mathbf{v}_i \Delta t_i - \frac{1}{2}\mathbf{g}\Delta t_i^2\right) - \Delta\mathbf{p}_i \\ \mathbf{R}_i^\top \left(\mathbf{v}_{i+1} - \mathbf{v}_i - \mathbf{g}\Delta t_i\right) - \Delta\mathbf{v}_i \\ \mathbf{b}_{i+1}^g - \mathbf{b}_i^g \\ \mathbf{b}_{i+1}^a - \mathbf{b}_i^a \end{bmatrix}, \\ E_{\mathrm{IMU}}(\mathcal{X}_{\mathcal{W}}) &= \frac{1}{2} \sum_{i=1}^{N-1} \left\|\mathbf{r}_{i,i+1}^I\right\|_{\mathbf{\Sigma}_{i,i+1}^{-1}}^2.\end{aligned} \tag{6}$$

Here, $\Delta t_i = \tau_{i+1} - \tau_i$, and $\Delta\mathbf{R}_i$, $\Delta\mathbf{p}_i$, and $\Delta\mathbf{v}_i$ are bias-corrected preintegrated increments. The gravity vector $\mathbf{g}$ remains fixed after initialization. Log maps a rotation to its three-dimensional rotation vector, and $\mathbf{\Sigma}_{i,i+1}$ is the residual covariance, with $\|\mathbf{r}\|_{\mathbf{\Sigma}^{-1}}^2 = \mathbf{r}^\top \mathbf{\Sigma}^{-1} \mathbf{r}$.

Metric position anchoring. Once optimized LiDAR–inertial poses are finalized, we interpolate them at supported keyframe timestamps and convert them into camera-pose anchors:

$$\mathbf{T}_j^{\mathrm{M}} = \mathbf{T}_{WW_{\mathrm{L}}}\, \mathbf{T}^{\mathrm{LI},\star}(t_j)\, \mathbf{T}_{BC}. \tag{7}$$

Here, $\mathbf{T}^{\mathrm{LI},\star}(t_j)$ is the interpolated IMU-to-world pose, $\mathbf{T}_{BC}$ maps camera coordinates to the IMU frame, and $\mathbf{T}_{WW_{\mathrm{L}}}$ rigidly aligns the LiDAR–inertial and visual world frames without scale fitting. Only keyframes bracketed by available finalized poses receive anchors. Their positions $\mathbf{p}_j^{\mathrm{M}}$ remain fixed during the subsequent visual position refinement, using the anchor prior defined in Section 3.2.2. Full pose anchors are retained for global correction and trajectory fusion; visual refinement does not modify the LiDAR–inertial window.

A.2.3 Observation-Supported Loop Correction

We obtain initial loop candidates and PGO proposals following (Liu et al., 2026). Each proposal is rigidly aligned to the current metric trajectory without scale fitting and converted to the camera-pose convention. Given current anchors $\mathbf{T}_i^{\mathrm{M}} = (\mathbf{R}_i^{\mathrm{M}}, \mathbf{p}_i^{\mathrm{M}})$ and aligned proposals $\widetilde{\mathbf{T}}_i^{\mathrm{M}}$, we independently control translation and rotation corrections:

$$\begin{aligned}\mathbf{p}_i^{\mathrm{M},\alpha} &= \mathbf{p}_i^{\mathrm{M}} + \alpha\left(\widetilde{\mathbf{p}}_i^{\mathrm{M}} - \mathbf{p}_i^{\mathrm{M}}\right), \\ \mathbf{R}_i^{\mathrm{M},\beta} &= \mathbf{R}_i^{\mathrm{M}}\, \mathrm{Exp}\Big(\beta\, \mathrm{Log}\Big((\mathbf{R}_i^{\mathrm{M}})^\top \widetilde{\mathbf{R}}_i^{\mathrm{M}}\Big)\Big).\end{aligned} \tag{8}$$

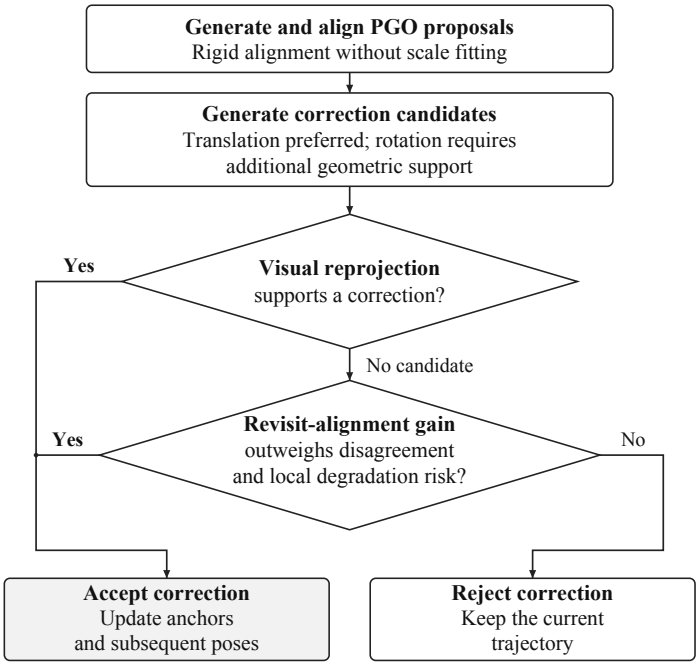

Figure 4: **Observation-supported loop correction.** Visual evidence is evaluated first, with geometric fallback when no visually supported candidate is selected. Rotation corrections require additional geometric support.

The gains α, β are drawn from a predefined discrete set $\mathcal{G} \subset [0, 1]$: zero preserves the current estimate, while one applies the full proposed correction.

Figure 4 summarizes candidate selection using available observations. Supported visual reprojection evidence is checked first. If no candidate is selected, geometric gains must outweigh disagreement across observation subsets and local degradation risk, with visual corroboration where available. Both branches favor translation-only corrections; rotation requires additional geometric benefit relative to its risk. Accepted corrections update the supported anchors and propagate to subsequent metric poses; rejection preserves the current accepted trajectory.

A.2.4 Metric-Anchored Visual Motion Fusion

We apply Equation 2 between consecutive accepted metric anchors with visual support, using the following interpolation and update rules:

(1) Interpolation. Metric anchors and visual endpoint poses are interpolated at the same sensor timestamps using linear position interpolation and shortest-arc quaternion SLERP. An interval is fused only after both anchor estimates become available.

(2) Anchor updates. Accepted loop corrections trigger recomputation of the affected fused poses using the updated anchors and the stored visual trajectory. The visual translation residual remains in world coordinates, while the visual rotation residual is composed with the interpolated metric orientation.

(3) Boundary handling. Before the right anchor becomes available, poses follow available LiDAR–IMU odometry relative to the latest accepted anchor and are revised upon finalization. A camera-only tail at sequence end follows visual relative motion from the last supported fused pose. Startup frames without metric support retain their initial pose estimates.

A.2.5 Optional Offline Hierarchical Pose Refinement

After streaming and loop correction, we initialize hierarchical LiDAR bundle adjustment (Liu et al., 2023) from the accepted camera trajectory, converted to scan-time IMU poses using calibrated extrinsics and coordinate transforms. Hierarchical optimization refines LiDAR alignment within and across submaps, and the resulting pose increments are transferred back to camera timestamps.

A.3 Geometry-Aware Gaussian Reconstruction Details

A.3.1 Streaming Gaussian Reconstruction

Representation and initialization. Following (Jiang et al., 2026), each triangle carries learnable vertices, opacity, and appearance features, and supports differentiable depth and normal rendering. Associated Gaussians model appearance, with their centers parameterized by the triangle barycenter plus learnable 3D offsets. For each mapping keyframe, positive differences between observed and rendered Laplacian-of-Gaussian responses identify unresolved image detail, while depth-adaptive spatial filtering suppresses redundant primitives. LiDAR-derived depth and normals initialize triangle positions and orientations. Associated Gaussians start at the triangle barycenter, with appearance initialized from RGB observations.

Rendering losses. With camera poses fixed during each mapping update, we jointly optimize geometry and appearance using

$$\begin{aligned}\mathcal{L}_{\text{stream}} &= \mathcal{L}_{\text{photo}} + \lambda_d \left\| \widehat{D}^{\text{T}} - D \right\|_1 + \lambda_n \left\| \widehat{\mathbf{N}}^{\text{T}} - \mathbf{N} \right\|_1 \\ &\quad + \lambda_{\text{inv}} \left\| (\widehat{D}^{\text{G}})^{-1} - D^{-1} \right\|_1 + \lambda_o \mathcal{L}_{\text{opacity}}, \\ \mathcal{L}_{\text{photo}} &= (1-\lambda_s) \left\| \widehat{\mathbf{I}}^{\text{G}} - \mathbf{I} \right\|_1 + \lambda_s \left[1 - \text{SSIM}(\widehat{\mathbf{I}}^{\text{G}}, \mathbf{I}) \right].\end{aligned} \tag{9}$$

Here, $\mathbf{I}$ is the observed RGB image, and $D, \mathbf{N}$ are LiDAR-derived depth and normal targets. Superscripts T and G denote triangle and Gaussian renderings, respectively. The norms denote mean absolute errors over valid pixels, and $\mathcal{L}_{\text{opacity}}$ regularizes triangle opacity following (Jiang et al., 2026). Depth and normal losses constrain triangle geometry, inverse-depth supervision constrains Gaussian placement, and photometric supervision refines appearance while also updating geometry through the triangle–Gaussian association. The λ coefficients balance these terms.

Pose updates. When a source-keyframe pose changes, we apply its rigid correction to the associated triangle vertices and Gaussian centers, rotating Gaussian offsets and orientations accordingly. This keeps the map consistent with the corrected camera trajectory.

CPU–GPU memory management. To reduce GPU memory usage as the scene grows, we offload inactive primitives and their optimizer states to CPU memory and reload them when their source keyframes become relevant. Triangle–Gaussian associations and optimizer states are preserved across transfers, allowing optimization to resume without reinitialization. Pose corrections update both CPU- and GPU-resident primitives to maintain consistency throughout the map.

A.3.2 Geometry-Aware Gaussian Refinement

Diagnostic evidence. We refine the exported Gaussians directly, keeping camera poses and exposure transforms fixed. Motion-compensated LiDAR projections provide depth and local normals for diagnosing regions with high RGB rendering error. For a Gaussian g, we evaluate

$$e_f(\mathbf{u}) = \frac{1}{3} \left\| \widehat{\mathbf{I}}_f^{\text{G}}(\mathbf{u}) - \mathbf{I}_f(\mathbf{u}) \right\|_1, \qquad E_g = \frac{1}{|\mathcal{O}_g|} \sum_{f \in \mathcal{O}_g} e_f(\mathbf{u}_{fg}), \tag{10}$$

where $\mathbf{u}_{fg}$ is the projected Gaussian center and $\mathcal{O}_g$ contains eligible training views with nearby LiDAR support, sufficient rendered opacity, and no foreground occlusion. Target positions and normals must agree across views. An observation is geometrically consistent when both its axial-depth error and point-to-plane distance are below the depth-dependent tolerance $\tau(z)$. It is inconsistent when its depth error lies between $2\tau(z)$ and an upper bound $d_{\max}$ and its RGB error exceeds τ_r. Let η_g^+ and η_g^- denote the corresponding fractions of eligible views.

Repair operations. We apply three repair operations based on rendering error, LiDAR depth agreement, and Gaussian coverage.

(1) Split oversized, geometrically consistent Gaussians. We select Gaussians with $E_g > \tau_s$, $\eta_g^+ > \eta_{\min}$, projected radius above $r_{\min}$, and largest scale $s_g > s_{\min}$. The corresponding principal axis $\mathbf{a}_g$ must be approximately tangential to the aggregated LiDAR normal $\mathbf{n}_g$, satisfying $|\mathbf{a}_g^\top \mathbf{n}_g| < \tau_{\tan}$. Each selected parent is replaced by two children:

$$\mathbf{c}_{g,\pm} = \mathbf{c}_g \pm \frac{s_g}{\sqrt{2}}\mathbf{a}_g, \qquad s_{g,\pm}^{(a)} = \frac{s_g}{\sqrt{2}}, \qquad \omega_{g,\pm} = \frac{\omega_g}{\sqrt{2}}, \tag{11}$$

where $s_{g,\pm}^{(a)}$ denotes the child scale along $\mathbf{a}_g$ and ω_g is the parent opacity. All other attributes are inherited.

(2) Replace Gaussians with persistent depth disagreement. We select Gaussians with $E_g > \tau_r$ and $\eta_g^- > \eta_{\min}$. Each center is projected onto LiDAR-supported tangent planes, and the projected positions are aggregated into a target $\mathbf{q}_g$. We reinitialize the Gaussian at $\mathbf{q}_g$, aligning its thin axis with the aggregated normal and initializing compact tangential scales and appearance from exposure-corrected training colors.

(3) Insert Gaussians in underrepresented regions. For a pixel with valid LiDAR depth z and rendering error $e_f(\mathbf{u}) > \tau_a$, insertion requires insufficient coverage and visibility, respectively:

$$\begin{aligned} &\widehat{A}_f(\mathbf{u}) < a_{\text{cov}} \quad \text{or} \quad \widehat{D}_f^{\text{G}}(\mathbf{u}) > z + 2\tau(z), \\ &\widehat{A}_f(\mathbf{u}) < a_{\text{empty}} \quad \text{or} \quad \widehat{D}_f^{\text{G}}(\mathbf{u}) \geq z - \tau(z). \end{aligned} \tag{12}$$

Both conditions must hold. Here, $\widehat{A}_f$ is accumulated rendered opacity; the second condition prevents insertion behind an already modeled foreground surface. Candidates must exhibit consistent normals and colors across multiple views. We retain the highest-error observation per qualifying spatial bin and reject candidates near existing Gaussian centers. LiDAR depth initializes position, local normals determine orientation, and training RGB observations initialize appearance.

Refinement objective. After repair, we optimize Gaussian centers, scales, rotations, opacities, and spherical-harmonic coefficients using

$$\begin{aligned} \mathcal{L}_{\text{refine}} &= \mathcal{L}_{\text{photo}} + \lambda_e \mathcal{L}_{\text{edge}} + \lambda_p \mathcal{L}_{\text{plane}}, \\ \mathcal{L}_{\text{plane}} &= \frac{1}{|\mathcal{A}|} \sum_{g \in \mathcal{A}} \text{SmoothL1}\left(\frac{\mathbf{n}_g^\top (\mathbf{c}_g - \mathbf{q}_g)}{\sigma_p} \right). \end{aligned} \tag{13}$$

The photometric term follows Equation 9. $\mathcal{L}_{\text{edge}}$ compares signed horizontal and vertical Sobel derivatives of image luminance at full and half resolution, using a Charbonnier penalty and detached residual-based weights. The set $\mathcal{A}$ includes newly created or reinitialized Gaussians with valid LiDAR targets, including supported split children, replacements, and insertions. Each target $(\mathbf{q}_g, \mathbf{n}_g)$ is held fixed during optimization, and σ_p sets the normal-displacement scale. The plane term is zero when $\mathcal{A}$ is empty. It constrains motion along the LiDAR normal while allowing tangential adjustment.

Repair schedule. Refinement runs for 15,000 steps. All three repair operations are applied every 1,000 steps from step 2,000 through 8,000. Unchanged primitives retain their Adam states, while split children, replacements, and insertions receive reset optimizer moments.

A.4 Data Acquisition and Benchmark

A.4.1 Sensor Suite and Robot Platform

Our data collection platform (Figure 5) integrates an industrial RGB camera (`SENSING TRD1-G2A GMSL2`), two Livox MID-360 LiDARs with built-in IMUs, and an onboard computer. Mounted on an AgileX mobile robot, the platform supports teleoperated data collection and online scene reconstruction. Each LiDAR has a horizontal field of view (FoV) of 360° and a vertical FoV of $-7°$ to $52°$. Following RISED (Jiang et al., 2025), we mount one LiDAR horizontally and the other at an incline to broaden spatial coverage while avoiding mutual occlusion. The horizontal

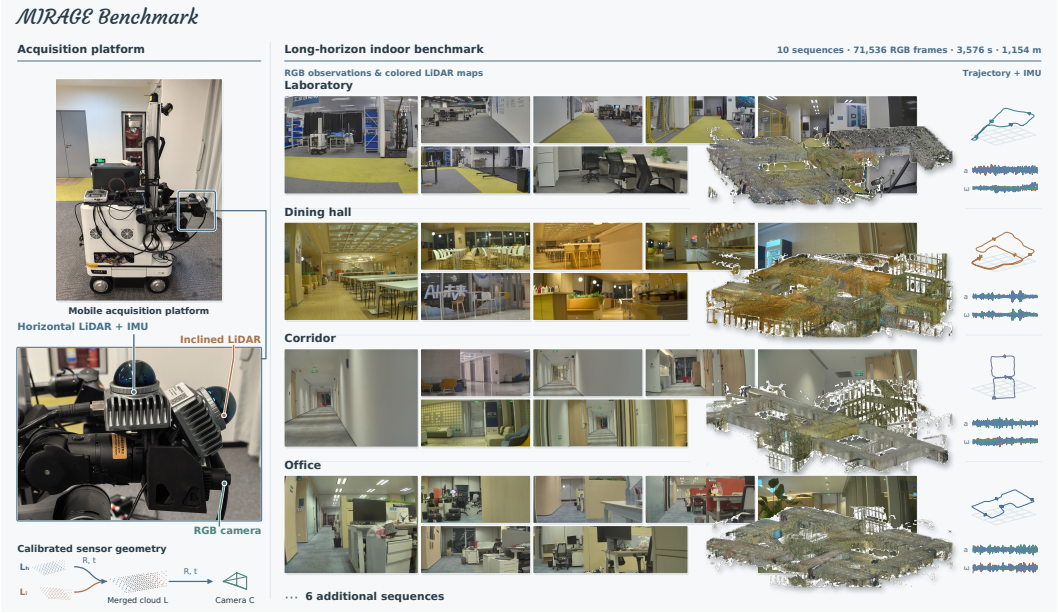

Figure 5: **Acquisition platform and dataset overview. Left:** the synchronized RGB–LiDAR–IMU platform, sensor layout, and calibrated sensor transforms. **Right:** RGB images, colored LiDAR maps, MIRAGE reference trajectories, and IMU signals from four representative sequences spanning laboratories, dining areas, corridors, and offices.

LiDAR captures distant structures to constrain pose estimation, while the inclined LiDAR densely samples nearby surfaces for detailed mapping. Inertial measurements are provided by the horizontal LiDAR's built-in IMU.

The camera is connected to an NVIDIA Thor device. It captures images at a resolution of 2880×1860 pixels, with a horizontal FoV of $64°$, a vertical FoV of $40°$, and an acquisition rate of $20\,\mathrm{Hz}$. Images and their associated timestamps are streamed from Thor to an onboard industrial computer through ZeroMQ. The industrial computer is equipped with an NVIDIA RTX 4090 GPU for SLAM and scene reconstruction.

A.4.2 Calibration and Time Synchronization

We estimate the inter-LiDAR extrinsics using GICP (Chen et al., 2022). Initialized from the assembly drawings, a shared rigid transform is jointly refined using point clouds collected in five distinct scenes with the platform stationary. LiDAR–camera extrinsics are estimated using FAST-Calib (Zheng & Zhang, 2026). We acquire paired images and LiDAR recordings of a calibration target at multiple positions, orientations, and distances to provide complementary geometric constraints. We qualitatively validate the estimated extrinsics by projecting RGB colors onto the LiDAR point cloud and visually inspecting the colorization for misalignment artifacts.

Time synchronization combines Precision Time Protocol (PTP) clock synchronization with external camera triggering. A Xingqi XQ500 synchronization unit provides the common time reference and camera trigger signals. The LiDAR clocks synchronize to the PTP reference over Ethernet. Each image is timestamped with its corresponding camera trigger time. We verify image timestamps by imaging a reference clock display and comparing the displayed times with the recorded timestamps, with observed discrepancies below $10\,\mathrm{ms}$.

A.4.3 Collection Protocol and Dataset Statistics

Our dataset comprises 10 sequences across four indoor scene categories: laboratories, dining areas, an office, and long corridors. Laboratories and dining areas contain numerous objects and reflec-

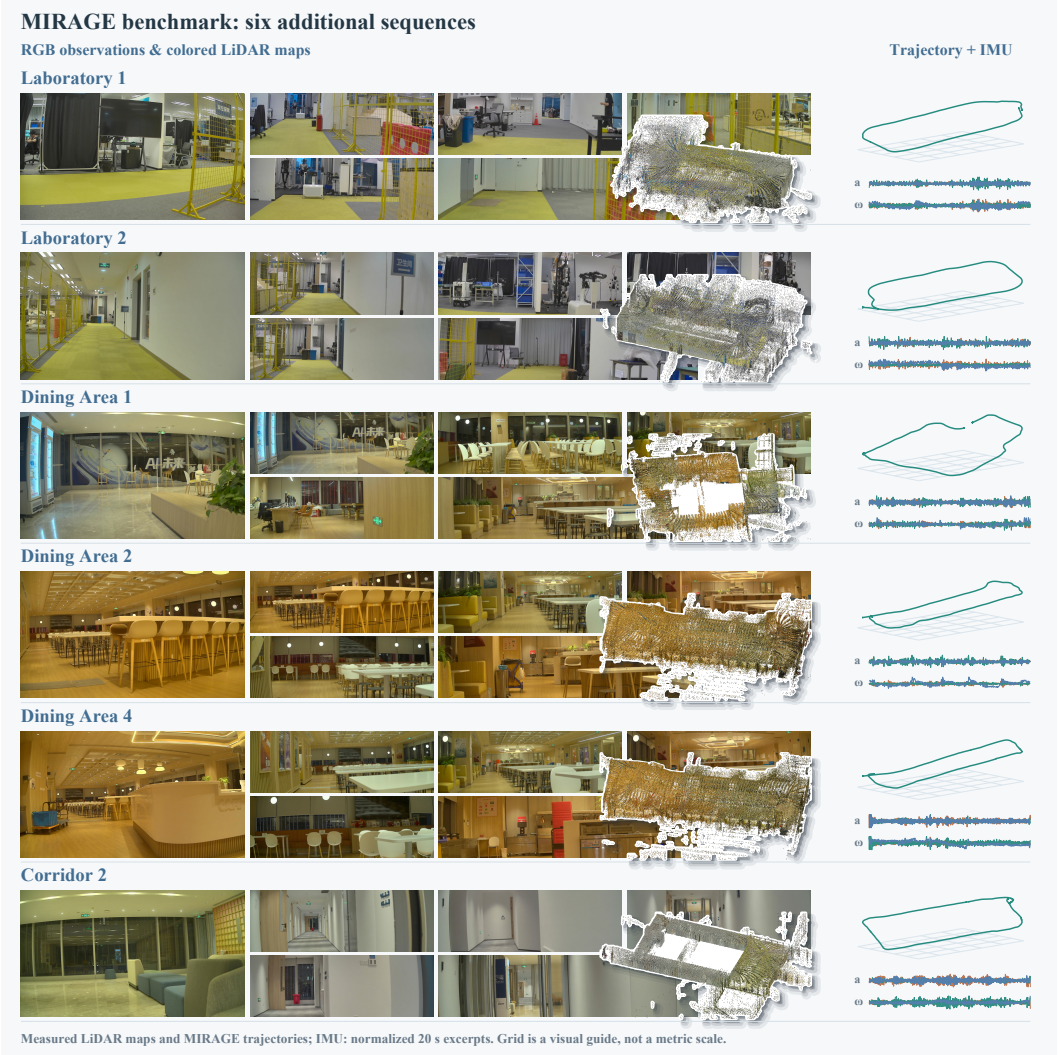

Figure 6: **Six additional MIRAGE benchmark sequences.** Complementing Figure 5, each row shows RGB images and colored LiDAR maps (left), alongside MIRAGE reference trajectories and IMU signals (right). IMU plots show normalized 20-second excerpts; trajectory grids are illustrative and do not indicate metric scale.

tive surfaces, challenging detailed geometry recovery and appearance modeling. Long, low-texture corridors challenge visual tracking and long-term trajectory consistency. We collected all sequences through low-speed teleoperation to limit motion blur and manually inspected the RGB images to remove blurred frames.

The dataset contains 71,536 RGB images at 2880×1860 resolution and 35,770 LiDAR frames, together with synchronized IMU measurements, spanning 59.6 minutes and approximately 1.15 km. Individual sequences range from 1.77 to 11.11 minutes, with trajectories approaching 200 m. Table 6 reports per-sequence statistics, with sequence names matching the benchmark evaluation. Figure 5 presents the acquisition platform and four representative sequences spanning all scene categories, while Figure 6 shows the remaining six sequences. Together, they illustrate scene appearance, LiDAR maps, reference trajectories, and inertial measurements across the complete dataset.

Table 6: Per-sequence acquisition statistics of the MIRAGE benchmark.

Sequence	# RGB Images	# LiDAR Frames	Duration (min)	Trajectory Length (m)
Laboratory 1	4,112	2,056	3.42	54.57
Laboratory 2	4,470	2,235	3.72	61.02
Laboratory 3	13,336	6,668	11.11	168.46
Dining Area 1	4,430	2,215	3.69	112.69
Dining Area 2	2,126	1,063	1.77	64.65
Dining Area 3	11,667	5,834	9.72	181.52
Dining Area 4	6,397	3,199	5.33	65.37
Corridor 1	11,729	5,865	9.77	165.39
Corridor 2	8,070	4,035	6.72	85.57
Office 1	5,199	2,600	4.33	195.15
Total	**71,536**	**35,770**	**59.60**	**1,154.40**

A.5 Implementation and Evaluation Protocols

Implementation. All experiments use NVIDIA RTX 4090 GPUs. LiDAR–inertial bundle adjustment uses a ten-scan sliding window. Gaussian refinement runs for 15,000 steps with camera poses and exposure transforms fixed, following the repair schedule in Appendix A.3.2. Offline 3DGS baselines are trained for 30,000 steps.

Evaluation protocols. Trajectory evaluation reports translational ATE RMSE after SE(3) alignment for metric methods and Sim(3) alignment for vision-only methods. Each trajectory is evaluated at timestamps with valid reference associations. Reported means weight sequences equally. Pose-stage comparisons use matched timestamps within each run; offline HBA results are reported separately rather than treated as a paired stage ablation.

Rendering evaluation holds out every eighth image from Gaussian optimization. On Oxford Spires, M2DGR, and INS, offline 3DGS uses fixed MIRAGE online poses. MIRAGE rendering results use the Gaussian maps after refinement, without HBA pose updates; Gaussian-LIC2 results use its online maps. Comparisons exclude Observatory 02, Church 03, and indoor_sq2 because Gaussian-LIC2 did not complete reconstruction.

On the MIRAGE benchmark, visual baselines use one-in-twelve sampling of the same preprocessed RGB sequences, without LiDAR, IMU, or reference-pose inputs. Each method supplies only its predicted poses to downstream 3DGS. These poses remain fixed during training, and Gaussians are initialized from training-view correspondences using a common triangulation procedure. The common-frame evaluation uses shared training and test images. MASt3R-SLAM is evaluated on registered keyframes, and its rendering results are excluded from common-frame rankings. Ours (reference) uses the same 3DGS procedure with MIRAGE reference poses. Stream PSNR reports the complete MIRAGE pipeline after refinement on its own held-out images, rather than the shared stride-12 test set.

For the visual ablation, camera poses are obtained by interpolating LiDAR–inertial poses to image timestamps and applying calibrated extrinsics; RGB supervision for Gaussian reconstruction is retained. Photometric and geometry-aware refinement start from the same streaming map and share the optimization budget, fixed poses and exposure, renderer, and test views. Photometric refinement uses only $\mathcal{L}_1$ and SSIM losses with a fixed primitive count.

A.6 Additional Experimental Results

Per-Sequence Pose Stages. Table 7 gives the per-sequence values underlying the online averages in Table 3(a). Offline (+HBA) averages use the same scene set and the unrounded source results for Table 1.

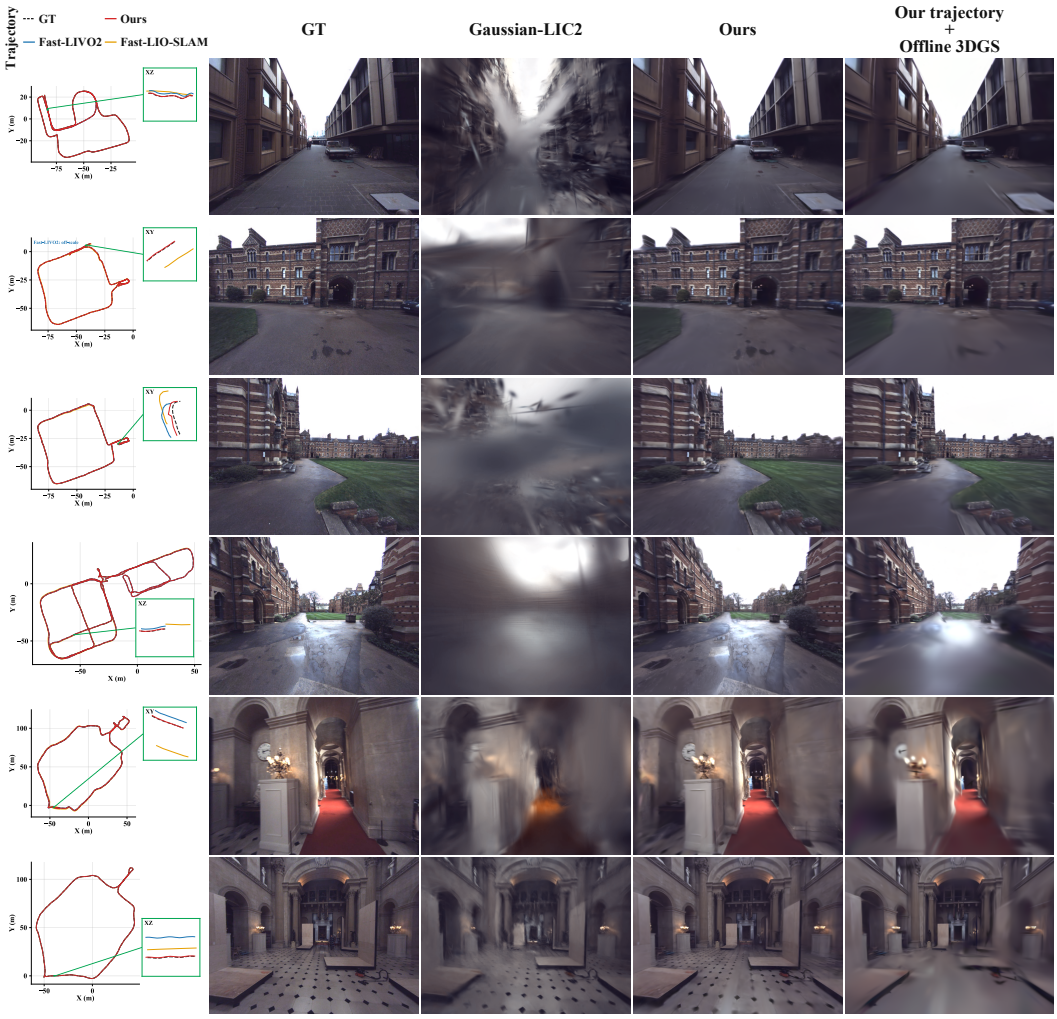

Figure 7: **Additional qualitative comparisons on Oxford Spires.** Rows, from top to bottom: Observatory 01, Keble 02, Keble 03, Keble 04, Blenheim 01, and Blenheim 02. Left: SE(3)-aligned XY trajectories, with green boxes and connectors indicating local views in the labeled planes at equal axis scales. Fast-LIVO2 is off-scale on Keble 02. Right: held-out ground-truth images, Gaussian-LIC2, MIRAGE, and offline 3DGS using MIRAGE trajectories.

Table 7: Stage-wise translational ATE RMSE (cm, ↓), with independent fixed-scale SE(3) alignment on common within-run timestamps. Tracking denotes saved visual inputs to metric refinement; Local and Global denote the no-loop and corrected camera trajectories.

	Oxford Spires								*M2DGR*			*INS*			
Stage	Observatory 01	Observatory 02	Keble 02	Keble 03	Keble 04	Blenheim 01	Blenheim 02	Church 03	room_01	room_02	room_03	indoor sq1	indoor sq2	indoor sq3	indoor sq4
Tracking	69.739	97.263	121.833	77.076	57.736	70.032	149.269	15.960	31.694	20.781	13.128	48.784	63.709	30.351	32.431
Local	4.242	4.695	5.083	7.162	10.725	14.654	14.473	1.239	1.828	1.926	1.971	13.148	5.468	5.172	4.433
Global	3.429	3.559	3.595	7.125	9.611	4.495	3.265	1.154	1.873	2.179	2.148	5.666	5.477	5.198	4.441

Figures 7 and 8 extend the trajectory and held-out rendering comparisons to all 12 sequences in Table 2. Observatory 02, Church 03, and indoor_sq2 are excluded because Gaussian-LIC2 failed on these sequences.

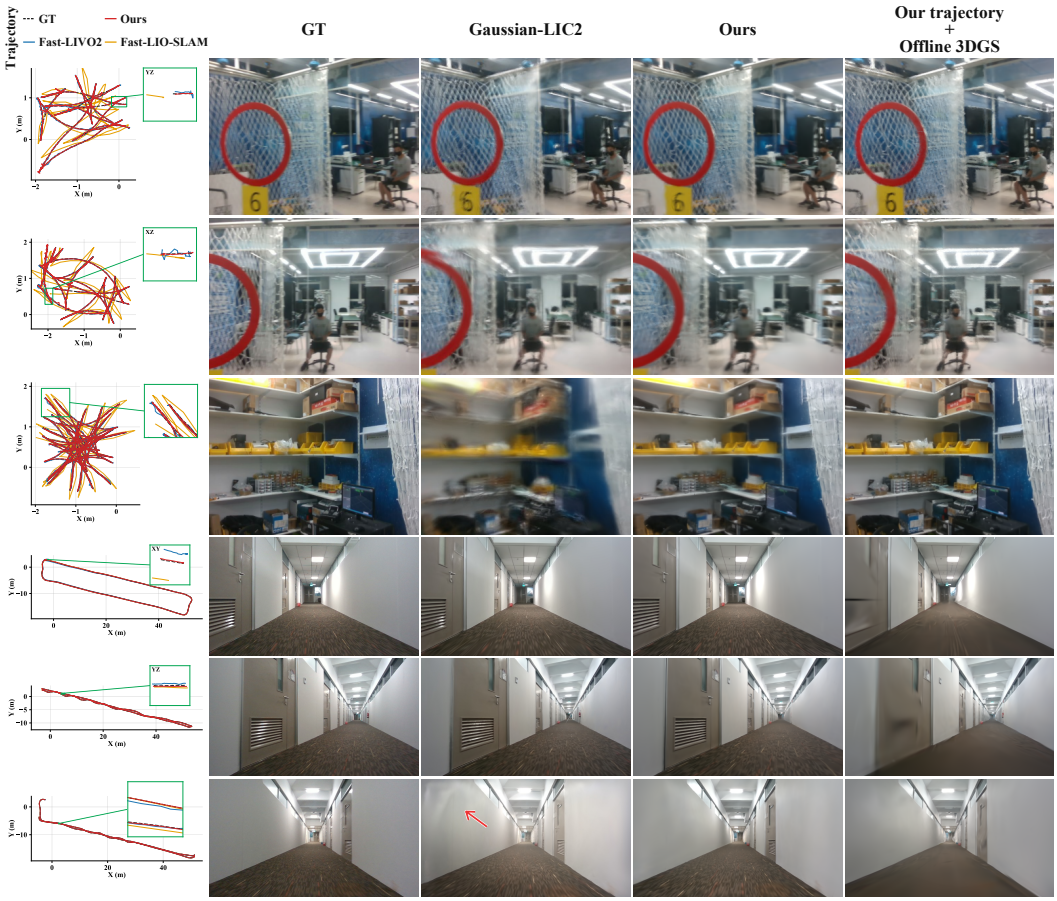

Figure 8: **Additional qualitative comparisons on M2DGR and INS.** Rows, from top to bottom: room_01, room_02, room_03, indoor_sq1, indoor_sq3, and indoor_sq4. Columns and green trajectory insets follow Figure 7. The arrow highlights a wall artifact in the Gaussian-LIC2 reconstruction of indoor_sq4.

A.7 LIMITATIONS

While MIRAGE achieves accurate metric pose estimation and high-fidelity reconstruction, several limitations remain. First, our benchmark focuses on static indoor scenes and does not assess the impact of dynamic objects on localization and reconstruction. Second, our data acquisition and reconstruction pipelines lack dedicated treatment of transparent and semi-transparent objects, which can compromise geometric accuracy and rendering quality. These limitations motivate future work on extending the benchmark to dynamic scenes and improving reconstruction of challenging materials.