ISR-SLAM: MULTI-AGENT 3DGS SLAM VIA INTER-SUBMAP REINFORCEMENT

Anonymous authors
Paper under double-blind review

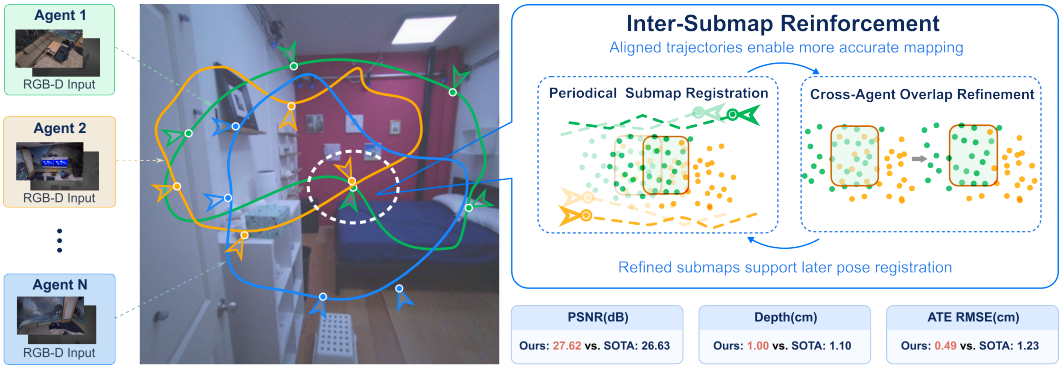

Figure 1: ISR-SLAM enables consistent multi-agent 3DGS SLAM through inter-submap reinforcement. Given RGB-D streams from multiple agents, ISR-SLAM continuously feeds optimized submap poses back to local mapping and refines overlapping Gaussian representations across agents. The resulting globally consistent Gaussian map supports accurate trajectory estimation, high-fidelity novel-view rendering, and geometrically consistent depth reconstruction.

ABSTRACT

Multi-agent simultaneous localization and mapping (SLAM) based on 3D Gaussian Splatting (3DGS) enables high-fidelity collaborative 3D reconstruction. A conventional pipeline partitions each agent's observations into short temporal segments and constructs a local submap with 3DGS for each segment. Reconstructed independently, the local 3DGS submaps are registered into a global map by optimizing inter-submap relative poses, after which the global 3DGS map is refined as a whole. However, this procedure breaks up submap registration and global map refinement. Consequently, defects in submaps can undermine the registration, which in turn hinders the global map refinement since submaps are incorrectly registered. With this insight, we propose **ISR-SLAM**, a multi-agent 3DGS SLAM framework that creates an online feedback loop between submap registration and global map refinement. As new submaps income, we implement periodical registration of all available submaps, ensuring subsequent tracking and mapping to proceed with the optimal registration up-to-date. Simultaneous to registration, we leverage cross-agent observation overlap to refine 3DGS submaps. The refined submaps, in turn, facilitates subsequent registration, forming a mutually reinforcing cycle. After merging all submaps, ISR-SLAM further refines the overlapped regions and submap boundaries to reduce residual inconsistencies. Extensive experiments on synthetic and real-world multi-agent datasets demonstrate that ISR-SLAM improves localization accuracy, reconstruction quality, and global map consistency significantly over existing methods.

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

Visual **S**imultaneous **L**ocalization **a**nd **M**apping (SLAM) jointly estimates camera motion and reconstructs the surrounding environment from sequential observations. The topic is fundamental to robotics, augmented reality, autonomous systems, and digital twins (Klein & Murray, 2007; Newcombe et al., 2011; Dai et al., 2017; Campos et al., 2021; Teed & Deng, 2021). In large and complex environments, multi-agent SLAM improves spatial coverage and exploration efficiency by allowing multiple agents to operate in parallel and share observations (Tian et al., 2022; Lajoie & Beltrame, 2024). Recent advances in 3D Gaussian Splatting (3DGS) (Kerbl et al., 2023) further introduces an efficient and expressive scene representation for collaborative SLAM, enabling high-quality rendering and dense reconstruction with minimal computational cost.

Conventionally, multi-agent 3DGS SLAM is designed as a submap-based collaborative pipeline (Yugay et al., 2025; Chen et al., 2026). Each agent first partitions its observation into short temporal segments, independently reconstructing a local 3DGS submap for each segment. After all local submaps are constructed, correspondences between overlapping submaps are established to provide geometric constraints for registering submaps in a shared coordinate system (Lajoie & Beltrame, 2024; Chen et al., 2026). Then, the registered submaps are integrated into a global 3DGS map, performing 3DGS refinement to reduce residual reconstruction inconsistencies (Yugay et al., 2025). In other words, existing pipeline proceeds as a sequence of separate stages: independent submap reconstruction first, followed by submap registration, and finally submap fusion and refinement.

Such a staged pipeline design results in a negative reinforcement, that defects in submaps can hinder registration, while conversely the incorrect registration degrades refinement of the fused global map. Submap registration and global map refinement are therefore natually entangled with each other: accurate submap registration facilitates global map refinement, and the refined 3DGS maps can better direct registration.

Motivated by this insight, we propose **ISR-SLAM**, a multi-agent 3D Gaussian SLAM framework that encourages cooperation between submap registration and 3DGS map refinement throughout mapping. As incoming submaps reconstructed, ISR-SLAM periodically corrects the registration of all available submaps, so that subsequent tracking and mapping can use the most accurate registration up-to-date. Simultaneously, ISR-SLAM optimizes the Gaussian primitives of 3DGS submaps according to inter-agent shared observations, instead of idling until the observation stream ends. By this design, we reduce reconstruction conflicts among agents ahead of global fusion, constructing positive reinforcement between submap registration and refinement.

Although our method refines 3DGS submaps before the final fusion, it cannot eliminate all inconsistencies throughout the scene, because the refinement simultaneous to registration is performed on computationally constrained local agents. Residual conflicts may persist after submap merging, particularly in overlapping regions and boundaries of neighboring submaps. ISR-SLAM therefore applies an additional refinement to the global map, which reconciles inconsistent representations in overlapping regions, improves continuity of neighboring submap boundaries, and finally performs a swift global adjustment. This final stage consolidates the local corrections introduced during online mapping into a coherent global Gaussian representation. To summarize the major contributions of this paper:

- We propose **ISR-SLAM**, a multi-agent 3D Gaussian SLAM framework that couples submap registration and refinement in an online manner during mapping, enabling them to positively reinforce each other.
- We develop an online submap registration strategy that periodically corrects the registration of all submaps as the stream incomes, enabling subsequent agent tracking and 3DGS mapping to utilize the most accurate registration available.
- We introduce an online cross-agent refinement mechanism that leverages reliable overlapping observations across agents to refine local 3DGS representations, and further develop an inter-submap overlap-aware global refinement strategy that resolves residual inconsistencies in inter-agent overlapping regions and neighboring submap boundaries.
- Extensive experiments on synthetic and real-world multi-agent datasets demonstrate the advantage of ISR-SLAM in localization accuracy, reconstruction quality, and global map consistency over existing methods.

2 Related Work

2.1 Gaussian SLAM

Recent advances in scene representation have enabled visual SLAM to move beyond geometric mapping toward dense and photorealistic reconstruction. Neural implicit representations, such as NeRF (Mildenhall et al., 2020), provide strong rendering quality but are less convenient for explicit geometric transformation and real-time map updates. Subsequent radiance-field representations further improved rendering quality and efficiency through multiscale sampling, tensor factorization, multiresolution hash encoding, and explicit acceleration structures (Barron et al., 2021; Chen et al., 2022; Müller et al., 2022; Yu et al., 2021). These scene representations have also been incorporated into dense neural SLAM systems (Sucar et al., 2021; Zhu et al., 2022; Yang et al., 2022; Johari et al., 2023; Wang et al., 2023).

More recently, 3DGS (Kerbl et al., 2023) represents the scene with explicit Gaussian primitives supporting differentiable rendering, well suited for both rendering and geometric manipulation. This representation has therefore been widely adopted in recent RGB-D SLAM systems (Yan et al., 2024; Keetha et al., 2024; Matsuki et al., 2024; Hu et al., 2024; Huang et al., 2024b; Peng et al., 2024). Sibling studies have also explored stronger geometric constraints. For example, 2D Gaussian Splatting improves surface accuracy by constraining Gaussian primitives to locally planar elements (Huang et al., 2024a). These methods have progressively improved tracking accuracy, reconstruction quality, and mapping efficiency. However, most of them are designed for a single agent, where tracking and map optimization are performed within a single trajectory.

2.2 Multi-Agent Gaussian SLAM

Multi-agent SLAM extends the mapping process to multiple agents by sharing observations and geometric constraints across different trajectories. Earlier systems mainly focus on cross-agent place recognition, loop closure, trajectory alignment, and global pose optimization under centralized or distributed architectures (Schmuck & Chli, 2017; 2019; Tian et al., 2022; Lajoie & Beltrame, 2024). Classical place recognition and loop-closure pipelines commonly build on local visual features such as SIFT and ORB, together with bag-of-words retrieval methods such as DBoW2 (Lowe, 2004; Rublee et al., 2011; Gálvez-López & Tardós, 2012). With the development of dense scene representations, collaborative SLAM has gradually moved from sparse geometric maps toward richer neural and point-based reconstruction (Hu et al., 2023).

More recently, 3DGS has been introduced into multi-agent SLAM to support efficient collaborative reconstruction. MAGiC-SLAM (Yugay et al., 2025) builds and aligns local Gaussian submaps from multiple agents before global fusion. Subsequent methods further improve cross-agent consistency, distributed mapping, and collaborative reconstruction quality (Xu et al., 2025; Chen et al., 2026). Meanwhile, OminiMAG-SLAM (Ding et al., 2026) brings submap-level pose optimization into online mapping, improving geometric consistency. However, the interaction between pose correction and cross-agent Gaussian representation refinement remains relatively less explored.

3 Method

As illustrated in Fig. 2, ISR-SLAM builds on a submap-based multi-agent 3DGS SLAM framework. Along with submap reconstruction, the system periodically optimizes the submap poses and feeds the resulting corrections back to the corresponding agents. Updated registration facilitates subsequent tracking and local 3DGS mapping to be more accurate. Inter-agent overlapping observations are simultaneously adopted to refine each local 3DGS submap, and in turn, benefits subsequent submap registration. Submap registration and refinement cooperate with each other tightly. Once the observation stream ends and all submaps are registered, the established overlap correspondences are further utilized to refine the integrated global map.

3.1 Preliminaries and Problem Setup

Multi-Agent 3DGS SLAM. Each agent receives an RGB-D stream and estimates its camera trajectory while incrementally reconstructing a local scene representation. We adopt 3DGS (Kerbl

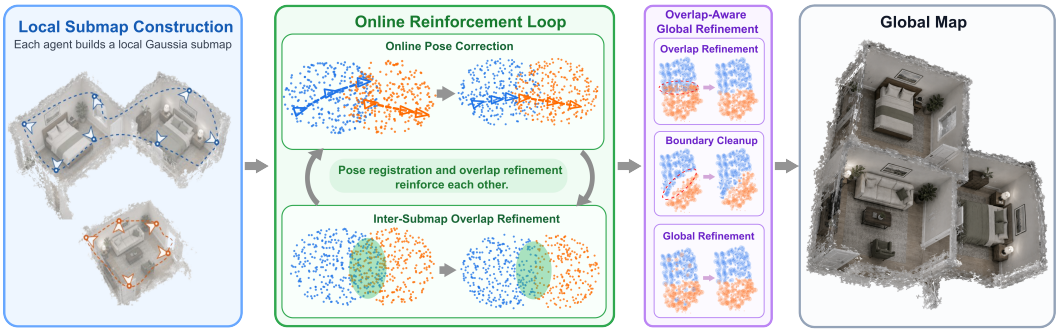

Figure 2: **Overall framework of ISR-SLAM.** Each agent constructs local Gaussian submaps from RGB-D observations. Registeredposes are fed back to local mapping and guide inter-agent overlap refinement, while refined submaps support subsequent tracking, forming an online reinforcement loop. After submap fusion, refinement focuses on shared regions and submap boundaries, followed by lightweight global updates to produce a consistent Gaussian map.

et al., 2023) to represent scene geometry and appearance using explicit Gaussian primitives. Given a Gaussian map $\mathcal{G}$ and a camera pose $\mathbf{T}$, differentiable rendering produces color, depth, and accumulated opacity:

$$(\hat{\mathbf{I}}, \hat{\mathbf{D}}, \hat{\mathbf{A}}) = \mathcal{R}(\mathcal{G}, \mathbf{T}), \tag{1}$$

where $\mathcal{R}(\cdot)$ denotes the Gaussian renderer. Gaussian SLAM (Keetha et al., 2024; Matsuki et al., 2024; Yan et al., 2024) typically alternates between tracking and mapping to process the input stream: tracking estimates the camera pose with the current map fixed, whereas mapping updates the 3DGS representation with selected RGB-D keyframes.

Local Gaussian Submaps. Following MAGiC-SLAM (Yugay et al., 2025), each agent partitions its RGB-D stream into short temporal segments, forming a local Gaussian submap $\mathcal{G}_i$ with an anchor pose $\mathbf{X}_i$ in the shared coordinate system. Completed submaps and their local trajectories are retained for inter-agent processing while the agent proceeds with a new incoming submap. Find more implementation details in Appendix A.1.

Submap Registration. When a new submap becomes available, the center server of agents retrieves and geometrically verifies historical submaps with potential scene overlap, following MAGiC-SLAM (Yugay et al., 2025). Validated relative transformations, together with temporal and loop constraints, form a growing submap-level pose graph.

Let $\mathbf{Z}_{ij}$ denote the measured relative transformation between connected submaps i and j. The server periodically optimizes the submap anchor poses as

$$\{\mathbf{X}_i^*\} = \arg\min_{\{\mathbf{X}_i\}} \sum_{(i,j)\in\mathcal{E}} \left\| \log\left(\mathbf{Z}_{ij}^{-1}\mathbf{X}_i^{-1}\mathbf{X}_j\right) \right\|_{\mathbf{\Omega}_{ij}}^2, \tag{2}$$

where $\mathcal{E}$ contains the valid submap constraints and $\mathbf{\Omega}_{ij}$ is the corresponding information matrix. For ISR-SLAM, as new submaps and constraints are added, the optimized poses evolve over time and provide the geometric corrections used by the online registration–refinement reinforcement. Find further details in Appendix A.1.

3.2 Online Registration–Refinement Reinforcement

ISR-SLAM feeds each registered submap back to the corresponding agent, allowing subsequent tracking and mapping to continue with updated context. Submap registration localizes each submap at correct relative position, but does not directly eliminate discrepancy of inter-submap overlap introduced by independent local submap reconstruction. Thus, we further refine the shared region of 3DGS submaps with their registration up-to-date. The refined local submaps remain in the system and participate in subsequent association and registration, forming an online reinforcement loop between submap registration and refinement.

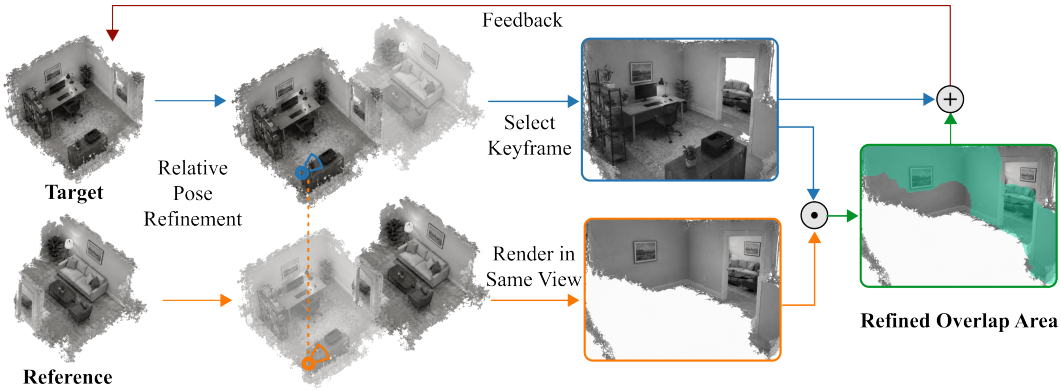

Figure 3: **Inter-submap overlap refinement.** Reliable overlap regions provide inter-submap supervision for updating the target Gaussian representation. The refined submap is reused in subsequent tracking and registration.

Online Registration Feedback

For submap i, denote its submap-level registration correction as:

$$\Delta \mathbf{T}_i = \mathbf{X}_i^* \mathbf{X}_i^{-1}. \tag{3}$$

The pose correction received during the construction of a submap is temporarily stored rather than applied immediately. When the current submap is completed, the agent applies the stored correction to the completed poses and submaps, and then initializes the next submap using the corrected pose.

To avoid discontinuities between adjacent trajectory segments, this correction is smoothly propagated over the preceding trajectory using quaternion SLERP for rotation and linear interpolation for translation, and is consistently applied to the associated keyframes and 3DGS. Details of the interpolation procedure are provided in Appendix A.2.

Subsequent tracking and mapping then proceed from the corrected trajectory and submap states. Loop and inter-agent constraints discovered during mapping can thereby influence local reconstruction before all submaps have been completed. The updated registration also provides a more reliable basis for the subsequent inter-submap overlap refinement.

Inter-Submap Overlap Refinement

Online registration feedback continually updates the relative position of local submaps, but 3DGS reconstructed independently by different agents may still disagree in overlapping observations. Even after pose alignment, discrepancies may remain in color, depth, and local geometry. We therefore use validated inter-agent overlapping pairs to refine the 3DGS of overlapping submaps.

As shown in Fig. 3, for each validated overlapping pair, one submap is selected as the trainable target, whereas the other remains fixed as the reference. Online registration provides an initial transformation $\mathbf{T}^0_{r\to t}$ from the reference submap to the selected target viewpoint. Although this transformation provides a globally consistent initial relation between the overlapping submaps, small patch-level misalignments may remain because of error in registration or reconstruction. We therefore refine the relative transformation before applying inter-submap supervision to 3DGS.

We introduce an incremental correction $\Delta \mathbf{T}_{r\to t}$ on top of the initial transformation. Rendering the fixed reference submap from the target viewpoint produces the inter-view reference:

$$\left(\mathbf{C}^{\text{ref}}, \mathbf{D}^{\text{ref}}, \mathbf{A}^{\text{ref}}\right) = \mathcal{R}\left(\mathcal{G}_r, \Delta \mathbf{T}_{r\to t} \mathbf{T}^0_{r\to t}\right). \tag{4}$$

We first refine the residual transformation while keeping both 3DGS maps fixed, and restrict inter-view supervision to reliable overlap pixels selected according to visibility and RGB-D consistency. The residual alignment objective and overlap-selection criteria are provided in Appendix A.3.

In the second stage, the reference submap and refined relative transformation are fixed, and only the target 3DGS map is updated. The refinement must incorporate complementary information from the reference view while preserving the target submap's consistency with its own observations and stabilizing local structure. We therefore optimize:

$$\mathcal{L}_{\text{refine}} = \lambda_{\text{local}}\mathcal{L}_{\text{local}} + \lambda_{\text{obs}}\mathcal{L}_{\text{obs}} + \lambda_{\text{inter}}\mathcal{L}_{\text{inter}}, \tag{5}$$

where $\mathcal{L}_{\text{local}}$ keeps the current target rendering close to the reference rendering saved before refinement, $\mathcal{L}_{\text{obs}}$ preserves consistency with the target agent's RGB-D observations, and $\mathcal{L}_{\text{inter}}$ enforces agreement with the corrected target reference over the reliable overlap region. Detailed loss definitions and optimization settings are provided in Appendix A.3.

The refined target 3DGS replaces its previous identity and remains available for subsequent submap registration. Thus, online submap registration facilitates submap refinement for 3DGS, whereas the refined 3DGS feeds back the registration for upcoming submaps. These two processes reinforce each other throughout an mapping procedure.

3.3 Overlap-Aware Global Refinement

Since online refinement only processes the overlapping pairs available during mapping, residual inconsistencies may remain after submap fusion, particularly in shared regions and neighboring submap transitions. We further exploit the validated inter-agent overlap relationships established during online mapping to reduce these residual inconsistencies. First, reliable shared regions are refined using the registration-corrected local submaps as references. Second, viewpoints around submap boundaries and overlap transitions are sampled more frequently to reduce local discontinuities. Finally, a swift global refinement with uniformly sampled training views stabilizes the overall geometry and appearance. Detailed sampling and optimization settings are provided in Appendix A.4.

4 Experiments

4.1 Experimental Setup

Implementation and Hardware. All methods are re-evaluated using their public implementations on the same server with four NVIDIA A40 GPUs and two Intel Xeon Gold 6342 CPUs.

Datasets. We evaluate ISR-SLAM on the synthetic ReplicaMultiagent (Straub et al., 2019; Hu et al., 2023) and the real-world AriaMultiagent (Pan et al., 2023; Yugay et al., 2025) benchmarks. Following MAGiC-SLAM, we select three continuous sequences with limited dynamic interference and stable observations from each room. Each sequence contains 500 frames for mapping, while 100 additional views per room are reserved for novel-view evaluation.

Baselines. We compare ISR-SLAM with MAGiC-SLAM (Yugay et al., 2025), CoMA-SLAM (Chen et al., 2026), and MAC-Ego3D (Xu et al., 2025). To ensure reproducible evaluation, we restrict the quantitative comparison to methods with publicly available implementations.

Metrics. Trajectory accuracy is evaluated using ATE RMSE (Sturm et al., 2012). Reconstruction quality is evaluated by PSNR, SSIM, LPIPS, and Depth L1 (Horé & Ziou, 2010; Wang et al., 2004; Zhang et al., 2018).

Efficiency Protocol. Following CoMA-SLAM (Chen et al., 2026), we report per-frame tracking and mapping time and final Gaussian-map storage over three runs. MAC-Ego3D is excluded because its asynchronous shared-map execution does not provide directly comparable runtime and storage boundaries.

4.2 Quantitative Results

Tracking Performance. Table 1 summarizes the trajectory estimation results on ReplicaMultiagent and AriaMultiagent. ISR-SLAM achieves the best or second-best average ATE in all evaluated

Table 1: Tracking accuracy on ReplicaMultiagent and AriaMultiagent (ATE RMSE in cm, ↓). Each entry lists the per-agent ATEs in order, followed by their average. Best results are bold; second-best are underlined.

Method	ReplicaMultiagent (Ag.1 / Ag.2 / Avg.)				AriaMultiagent (Ag.1 / Ag.2 / Ag.3 / Avg.)	
	Apart0	Apart1	Apart2	Office	Room0	Room1
MAGiC-SLAM	0.16 / 0.20 / 0.18	0.24 / 0.30 / 0.27	0.48 / 0.23 / 0.36	0.25 / 0.20 / 0.23	0.74 / 1.16 / 1.68 / 1.19	0.95 / 0.54 / 0.46 / 0.65
CoMA-SLAM	1.35 / 0.82 / 1.09	1.97 / 1.45 / 1.71	1.93 / 1.49 / 1.71	0.16 / 0.30 / 0.23	0.99 / 0.96 / 1.47 / 1.14	1.74 / 0.62 / 1.32 / 1.23
MAC-Ego3D	**0.11** / 0.13 / 0.12	**0.18** / **0.20** / **0.19**	**0.10** / **0.11** / **0.11**	0.19 / **0.10** / 0.15	1.67 / **0.77** / **1.08** / 1.17	0.89 / **0.25** / **0.33** / **0.49**
Ours	0.12 / **0.10** / **0.11**	0.19 / 0.22 / 0.21	0.20 / 0.19 / 0.19	**0.11** / 0.12 / **0.11**	**0.72** / 0.89 / 1.20 / **0.94**	**0.84** / 0.29 / 0.35 / **0.49**

Table 2: Rendering quality on ReplicaMultiagent and AriaMultiagent. Best results are bold; second-best are underlined.

Method	Metric	ReplicaMultiagent				AriaMultiagent			
						Novel Views		Training Views	
		Apart0	Apart1	Apart2	Office	Room0	Room1	Room0	Room1
MAGiC-SLAM	PSNR [dB] ↑	37.39	29.91	30.61	39.28	23.40	21.87	24.17	26.15
	SSIM ↑	0.98	0.95	0.96	**0.99**	0.88	0.86	0.90	0.93
	LPIPS ↓	0.09	0.18	0.18	0.06	0.25	0.22	0.21	0.17
	Depth L1 [cm] ↓	0.64	3.02	1.01	0.42	1.33	**5.10**	2.00	1.30
CoMA-SLAM	PSNR [dB] ↑	38.99	30.86	28.26	**42.96**	24.11	21.53	25.49	26.63
	SSIM ↑	**0.99**	0.96	0.91	**0.99**	**0.91**	0.85	**0.93**	0.94
	LPIPS ↓	0.05	0.12	0.17	**0.02**	**0.21**	0.20	**0.18**	0.12
	Depth L1 [cm] ↓	**0.37**	**1.57**	1.60	**0.20**	**0.90**	**5.10**	**1.30**	1.10
MAC-Ego3D	PSNR [dB] ↑	**43.38**	**33.84**	**38.63**	42.34	22.91	21.12	23.31	26.29
	SSIM ↑	0.98	0.96	0.98	0.98	0.89	0.83	0.89	0.93
	LPIPS ↓	**0.04**	**0.07**	**0.06**	0.03	0.26	0.21	0.23	0.14
	Depth L1 [cm] ↓	0.86	3.99	1.00	0.88	2.12	5.36	2.42	1.48
Ours	PSNR [dB] ↑	42.10	33.10	35.07	42.30	**25.51**	**22.35**	**25.69**	**27.62**
	SSIM ↑	**0.99**	**0.97**	**0.99**	**0.99**	0.89	**0.87**	0.91	**0.95**
	LPIPS ↓	**0.04**	0.12	0.09	0.03	0.22	**0.18**	0.19	**0.11**
	Depth L1 [cm] ↓	0.40	2.30	**0.60**	0.30	1.70	5.40	2.20	**1.00**

scenes. It maintains stable localization on both ReplicaMultiagent and the more challenging real-world AriaMultiagent, indicating that online geometric feedback effectively limits trajectory drift across different multi-agent environments.

Rendering and Reconstruction Performance. Table 2 reports rendering and reconstruction quality on both benchmarks. On ReplicaMultiagent, ISR-SLAM consistently achieves strong structural and geometric reconstruction quality, with leading SSIM and competitive LPIPS and Depth L1 results. On AriaMultiagent, ISR-SLAM shows particularly strong performance despite the challenges of real-world RGB-D data, including sensor noise, incomplete depth observations, and appearance variations. It achieves the best PSNR across all four evaluated views, including both novel and training views, with especially clear gains on the novel views. It also maintains competitive SSIM and LPIPS, indicating that the improvement is not limited to pixel-level reconstruction quality but extends to perceptual and structural consistency. Depth L1 also remains competitive with the best result on Training-R1 and only modest differences from the strongest baselines on the other views.

4.3 Ablation and Analysis

All ablation experiments are conducted on the Office scene of ReplicaMultiagent.

Analysis of Inter-Submap Reinforcement. We evaluate three variants: *w/o Online Pose Feedback* postpones pose registration until local mapping is completed; *w/o Inter-Agent Overlap Refinement* disables inter-agent overlap refinement; and *w/o Reinforcement Loop* prevents refined submaps

Table 3: Ablation of inter-submap reinforcement on ReplicaMultiagent Office.

Method	ATE [cm] ↓	PSNR [dB] ↑	Depth L1 [cm] ↓
w/o Online Pose Feedback	0.19	39.84	0.42
w/o Inter-Agent Overlap Refinement	0.16	38.31	0.46
w/o Reinforcement Loop	0.15	39.12	0.38
Ours (Full)	**0.12**	**42.30**	**0.30**

Table 4: Ablation of overlap-aware global refinement on ReplicaMultiagent Office.

Method	PSNR ↑	SSIM ↑	LPIPS ↓	Depth L1 ↓
Merged Map	28.94	0.90	0.24	4.70
+ Overlap-Aware Refinement	38.41	0.98	0.06	0.80
+ Boundary-Focused Sampling	40.87	0.99	0.04	0.50
Ours (Full)	**42.30**	**0.99**	**0.03**	**0.30**

Table 5: Efficiency and Gaussian-map storage comparison on ReplicaMultiagent Office. Best results are bold; second-best are underlined.

Method	Tracking/Frame ↓	Mapping/Frame ↓	Memory ↓	Gaussian Count ↓
MAGiC-SLAM	**0.62 ± 0.03 s**	11.27 ± 0.31 s	46.49 ± 0.09 MB	830,115.67 ± 1,540.51
CoMA-SLAM	0.99 ± 0.05 s	13.09 ± 0.36 s	**40.36 ± 0.08 MB**	**561,316.31 ± 1,041.68**
Ours	**0.62 ± 0.01 s**	**11.21 ± 0.29 s**	45.06 ± 0.18 MB	–

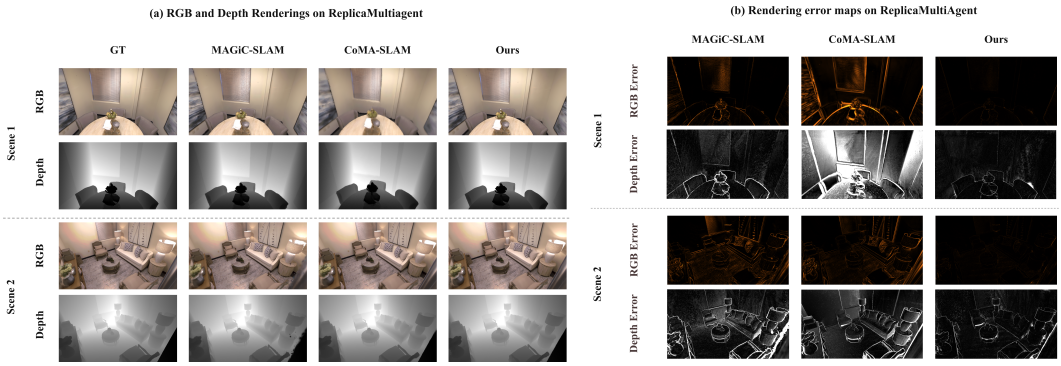

Figure 4: **Qualitative rendering results on ReplicaMultiagent.** (a) RGB and depth renderings of two scenes compared with ground truth. (b) ISR-SLAM exhibits fewer high-error regions around object contours and depth boundaries.

from participating in subsequent association and registration. Table 3 shows that removing either online pose feedback or inter-agent refinement degrades performance. Preventing refined submaps from participating in later association also increases tracking error, validating the reinforcement loop between online pose registration and overlap refinement.

Effect of Overlap-Aware Global Refinement. Table 4 evaluates the three refinement stages cumulatively. Starting from the directly merged map, overlap-aware refinement provides the largest gain by exploiting inter-agent observations in shared regions. Boundary-focused sampling further improves the continuity around submap transitions, while the final global updates provide additional improvements in the overall appearance and geometric consistency of the merged representation. These results indicate that the three stages address complementary residual inconsistencies at different spatial scales.

4.4 Efficiency Analysis

Table 5 shows that ISR-SLAM maintains tracking and mapping costs comparable to MAGiC-SLAM while using slightly less Gaussian-map storage. These results indicate that the proposed online collaboration introduces limited overhead to the local SLAM pipeline. The measured per-frame local tracking and mapping costs remain comparable to MAGiC-SLAM despite the additional collaborative processing.

4.5 Qualitative Results

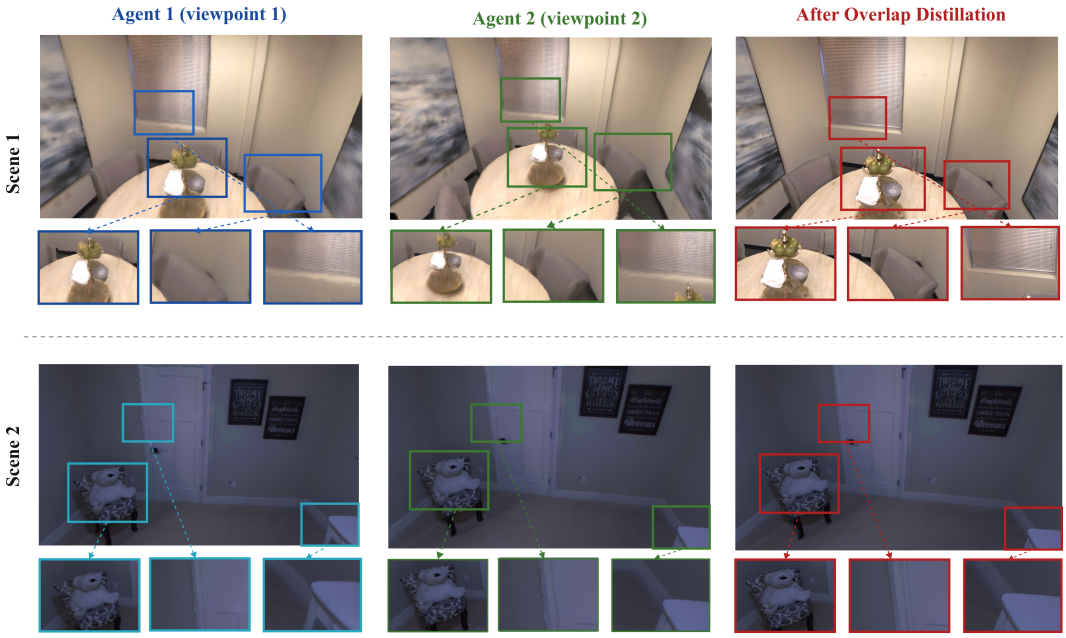

Figure 5: **Qualitative results of inter-submap overlap refinement.** The first two columns show shared regions from two agents' viewpoints, while the third shows the refined results. Enlarged crops highlight improved texture details, clearer object contours, and reduced artifacts around structural boundaries across two scenes.

RGB and Depth Rendering. Figure 4 shows the RGB, depth, and corresponding error maps on ReplicaMultiagent. Brighter error-map regions indicate larger deviations from ground truth. ISR-SLAM provides more stable reconstruction of local textures, object contours, and depth boundaries while preserving the overall scene structure.

Overlap-Region Refinement. Figure 5 shows that overlap-aware refinement improves local details and structural continuity in shared regions. Compared with the independently reconstructed submaps, the refined results exhibit clearer textures and object contours, while reducing visible artifacts around shared structures and boundaries. The improvement is particularly evident in the enlarged regions, where inconsistent details across submaps are substantially reduced. These qualitative observations are consistent with the gains reported in the global-refinement ablation results.

5 Conclusion

We present ISR-SLAM, a multi-agent 3DGS SLAM framework that introduces inter-submap reinforcement to couple submap registration and 3DGS refinement throughout the reconstruction process. ISR-SLAM continuously feeds registered submap poses back to local mapping, reducing the influence of accumulated pose errors on subsequent 3DGS reconstruction. At the same time, inter-agent overlap refinement exploits shared observations to improve the consistency of Gaussian representations across independently constructed submaps. The refined submaps are then used for subsequent tracking and registration, forming a bidirectional reinforcement loop between online pose registration and inter-agent overlap refinement. After submap fusion, an overlap-aware global refinement further addresses residual inconsistencies in shared regions and around submap boundaries. Experiments on ReplicaMultiagent and AriaMultiagent demonstrate that ISR-SLAM achieves accurate trajectory estimation while maintaining high-quality appearance and geometric reconstruction. Ablation studies further verify the complementary contributions of online pose feedback, inter-agent overlap refinement, and the reinforcement loop, while the global refinement stages effectively reduce residual artifacts after fusion. Overall, ISR-SLAM provides an efficient approach for improving the geometric and representational consistency of multi-agent 3DGS SLAM.

AI USE STATEMENT

In this work, we used generative AI tools for only word refinement. We have not used generative AI tools for other tasks requiring disclosure, including the development of theoretical models or conceptual frameworks, the formulation of mathematical claims or proofs, the proposal or modification of hypotheses, research method or experimental design, interpretation of experimental results, implementation of the proposed method, or the creation or modification of scientific figures. We have reviewed all AI-assisted work and manually verified the final content for accuracy and consistency. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We provide the main algorithmic formulation and experimental setup in the main paper. Additional implementation details, including submap construction, online pose registration scheduling, smooth pose feedback, inter-submap overlap refinement, and global refinement settings, are provided in Appendix A. All baselines are evaluated using their public implementations under the same hardware setting.

REFERENCES

Jonathan T. Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P. Srinivasan. Mip-NeRF: A multiscale representation for anti-aliasing neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5855–5864, 2021.

Carlos Campos, Richard Elvira, Juan J. Gómez Rodríguez, José M. M. Montiel, and Juan D. Tardós. ORB-SLAM3: An accurate open-source library for visual, visual-inertial, and multi-map SLAM. *IEEE Transactions on Robotics*, 37(6):1874–1890, 2021. doi: 10.1109/TRO.2021.3075644.

Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. TensoRF: Tensorial radiance fields. In *Proceedings of the European Conference on Computer Vision*, pp. 333–350, 2022.

Lin Chen, Yongxin Su, Jvboxi Wang, Pengcheng Han, Zhenyu Xia, Shuhui Bu, Kun Li, Boni Hu, Shengqi Meng, and Guangming Wang. CoMA-SLAM: Collaborative multi-agent gaussian SLAM with geometric consistency. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(4): 2922–2929, 2026. doi: 10.1609/aaai.v40i4.37283.

Yang Chen and Gérard Medioni. Object modelling by registration of multiple range images. *Image and Vision Computing*, 10(3):145–155, 1992.

Angela Dai, Matthias Nießner, Michael Zollhöfer, Shahram Izadi, and Christian Theobalt. BundleFusion: Real-time globally consistent 3d reconstruction using on-the-fly surface reintegration. *ACM Transactions on Graphics*, 36(3):24:1–24:18, 2017. doi: 10.1145/3054739.

Leqian Ding, Caibo Li, Yu Guo, and Fei Wang. OminiMAG-SLAM: Unified online dual graph optimization for multi-agent gaussian SLAM. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Findings*, pp. 1431–1440, 2026.

Dorian Gálvez-López and Juan D. Tardós. Bags of binary words for fast place recognition in image sequences. *IEEE Transactions on Robotics*, 28(5):1188–1197, 2012. doi: 10.1109/TRO.2012.2197158.

Alain Horé and Djemel Ziou. Image quality metrics: PSNR vs. SSIM. In *Proceedings of the International Conference on Pattern Recognition*, pp. 2366–2369, 2010.

Jiarui Hu, Mao Mao, Hujun Bao, Guofeng Zhang, and Zhaopeng Cui. CP-SLAM: Collaborative neural point-based SLAM system. In *Advances in Neural Information Processing Systems*, volume 36, pp. 39429–39442, 2023.

Jiarui Hu, Xianhao Chen, Boyin Feng, Guanglin Li, Liangjing Yang, Hujun Bao, Guofeng Zhang, and Zhaopeng Cui. CG-SLAM: Efficient dense RGB-D SLAM in a consistent uncertainty-aware 3d gaussian field. In *Proceedings of the European Conference on Computer Vision*, pp. 93–112, 2024.

Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2d gaussian splatting for geometrically accurate radiance fields. In *ACM SIGGRAPH 2024 Conference Papers*, pp. 1–11, 2024a. doi: 10.1145/3641519.3657428.

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*, pp. 21584–21593, 2024b. doi: 10.1109/CVPR52733.2024.02039.

Mohammad Mahdi Johari, Camilla Carta, and François Fleuret. ESLAM: Efficient dense SLAM system based on hybrid representation of signed distance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17408–17419, 2023.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 7(3):535–547, 2021. doi: 10.1109/TBDATA.2019.2921572.

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*, pp. 21357–21366, 2024.

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

Georg Klein and David Murray. Parallel tracking and mapping for small AR workspaces. In *Proceedings of the IEEE and ACM International Symposium on Mixed and Augmented Reality*, pp. 225–234, 2007.

Pierre-Yves Lajoie and Giovanni Beltrame. Swarm-SLAM: Sparse decentralized collaborative simultaneous localization and mapping framework for multi-robot systems. *IEEE Robotics and Automation Letters*, 9(1):475–482, 2024.

David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

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*, pp. 18039–18048, 2024. doi: 10.1109/CVPR52733.2024.01708.

Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. In *Proceedings of the European Conference on Computer Vision*, pp. 405–421, 2020.

Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics*, 41(4):102:1–102:15, 2022. doi: 10.1145/3528223.3530127.

Richard A. Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J. Davison, Pushmeet Kohli, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. KinectFusion: Real-time dense surface mapping and tracking. In *Proceedings of the IEEE International Symposium on Mixed and Augmented Reality*, pp. 127–136, 2011.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024.

Xiaqing Pan, Nicholas Charron, Yongqian Yang, Scott Peters, Thomas Whelan, Chen Kong, Omkar Parkhi, Richard Newcombe, and Yuheng (Carl) Ren. Aria digital twin: A new benchmark dataset for egocentric 3d machine perception. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 20133–20143, 2023.

Zhexi Peng, Tianjia Shao, Yong Liu, Jingke Zhou, Yin Yang, Jingdong Wang, and Kun Zhou. RTG-SLAM: Real-time 3d reconstruction at scale using gaussian splatting. In *ACM SIGGRAPH 2024 Conference Papers*, pp. 1–11, 2024. doi: 10.1145/3641519.3657455.

Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. ORB: An efficient alternative to SIFT or SURF. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2564–2571, 2011.

Patrik Schmuck and Margarita Chli. Multi-uav collaborative monocular SLAM. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3863–3870, 2017. doi: 10.1109/ICRA.2017.7989445.

Patrik Schmuck and Margarita Chli. CCM-SLAM: Robust and efficient centralized collaborative monocular simultaneous localization and mapping for robotic teams. *Journal of Field Robotics*, 36(4):763–781, 2019.

Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J. Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, Anton Clarkson, Mingfei Yan, Brian Budge, Yajie Yan, Xiaqing Pan, June Yon, Yuyang Zou, Kimberly Leon, Nigel Carter, Jesus Briales, Tyler Gillingham, Elias Mueggler, Luis Pesqueira, Manolis Savva, Dhruv Batra, Hauke Strasdat, Renzo De Nardi, Michael Goesele, Steven Lovegrove, and Richard Newcombe. The replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019.

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

Edgar Sucar, Shikun Liu, Joseph Ortiz, and Andrew J. Davison. iMAP: Implicit mapping and positioning in real-time. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6229–6238, 2021.

Zachary Teed and Jia Deng. DROID-SLAM: Deep visual SLAM for monocular, stereo, and RGB-D cameras. In *Advances in Neural Information Processing Systems*, volume 34, pp. 16558–16569, 2021.

Yulun Tian, Yun Chang, Fernando Herrera Arias, Carlos Nieto-Granda, Jonathan P. How, and Luca Carlone. Kimera-Multi: Robust, distributed, dense metric-semantic SLAM for multi-robot systems. *IEEE Transactions on Robotics*, 38(4):2022–2038, 2022.

Hengyi Wang, Jingwen Wang, and Lourdes Agapito. Co-SLAM: Joint coordinate and sparse parametric encodings for neural real-time SLAM. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13293–13302, 2023.

Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4): 600–612, 2004.

Xiaohao Xu, Feng Xue, Shibo Zhao, Yike Pan, Sebastian Scherer, and Xiaonan Huang. MAC-Ego3D: Multi-agent gaussian consensus for real-time collaborative ego-motion and photorealistic 3d reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 854–863, 2025.

Chi Yan, Delin Qu, Dan Xu, Bin Zhao, Zhigang Wang, Dong Wang, and Xuelong Li. GS-SLAM: Dense visual SLAM with 3d gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19595–19604, 2024.

Xingrui Yang, Hai Li, Hongjia Zhai, Yuhang Ming, Yuqian Liu, and Guofeng Zhang. Vox-Fusion: Dense tracking and mapping with voxel-based neural implicit representation. In *Proceedings of the IEEE International Symposium on Mixed and Augmented Reality*, pp. 499–507, 2022.

Alex Yu, Ruilong Li, Matthew Tancik, Hao Li, Ren Ng, and Angjoo Kanazawa. PlenOctrees for real-time rendering of neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5752–5761, 2021.

Vladimir Yugay, Theo Gevers, and Martin R. Oswald. MAGiC-SLAM: Multi-agent gaussian globally consistent SLAM. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6741–6750, 2025.

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*, pp. 586–595, 2018.

Zihan Zhu, Songyou Peng, Viktor Larsson, Weiwei Xu, Hujun Bao, Zhaopeng Cui, Martin R. Oswald, and Marc Pollefeys. NICE-SLAM: Neural implicit scalable encoding for SLAM. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12786–12796, 2022.

A Additional Implementation Details

This section provides additional implementation details of ISR-SLAM that are omitted from the main paper for brevity. We focus on the scheduling of local submaps and online pose registration, the implementation of online pose feedback and overlap refinement, and the staged global refinement.

A.1 Submap Construction, Registration, and Online PGO Scheduling

ISR-SLAM constructs local Gaussian submaps at fixed temporal intervals. On ReplicaMultiagent, a new submap is initialized every 50 frames, whereas on AriaMultiagent the interval is 20 frames. For both datasets, adjacent submaps retain a 10-frame overlap window around their temporal boundaries.

Following MAGiC-SLAM (Yugay et al., 2025), candidate overlapping submaps are retrieved using DINOv2 descriptors (Oquab et al., 2024) indexed with FAISS (Johnson et al., 2021). The retrieved candidates are geometrically verified using point-to-plane ICP (Chen & Medioni, 1992), and only validated relative transformations are inserted into the submap-level pose graph. Candidate registrations with insufficient registration fitness or excessive inlier RMSE are rejected before pose-graph construction.

Online submap registration is triggered by newly received submaps. A new registration update is initiated once the server has received at least eight new submaps in total from all agents since the previous registration round. The registration update is executed only when sufficient graph constraints are available. In addition, every agent must have produced at least one submap, and at least one valid loop constraint must remain after filtering. The resulting pose corrections are applied only at submap boundaries.

A.2 Smooth Online Pose Feedback

For submap i, let $\mathbf{X}_i$ denote its current anchor pose and $\mathbf{X}_i^*$ the updated anchor pose produced by the latest online pose registration update. The corresponding submap-level correction is

$$\Delta\mathbf{T}_i = \mathbf{X}_i^* \mathbf{X}_i^{-1}. \tag{6}$$

The correction is cached until the active submap reaches the next submap boundary. To avoid discontinuities between adjacent trajectory segments, the correction is smoothly propagated over the preceding segment. Let $\mathbf{R}_\Delta$ and $\mathbf{t}_\Delta$ denote the rotational and translational components of $\Delta\mathbf{T}_i$. For frame k between the previous correction boundary t_{last} and the current boundary t_{curr}, we define

$$\beta_k = \frac{k - t_{\text{last}}}{t_{\text{curr}} - t_{\text{last}}}, \tag{7}$$

and

$$\Delta\mathbf{T}_k^{\text{frame}} = \begin{bmatrix} \text{SLERP}(\mathbf{I}, \mathbf{R}_\Delta; \beta_k) & \beta_k \mathbf{t}_\Delta \\ \mathbf{0}^\top & 1 \end{bmatrix}, \qquad \mathbf{T}_{wc,k}^* = \Delta\mathbf{T}_k^{\text{frame}} \mathbf{T}_{wc,k}. \tag{8}$$

The rotational correction is interpolated using spherical linear interpolation, while the translation is interpolated linearly. The same correction is propagated to the associated historical keyframes and completed Gaussian submaps, including the positions and orientations of their Gaussian primitives. This ensures that subsequent tracking and mapping continue from mutually consistent trajectory and map states.

A.3 Inter-submap Overlap Refinement Details

Reliable overlap selection. Only pixels satisfying visibility and RGB-D consistency constraints are used for inter-submap overlap refinement. We use

$$\tau_\alpha = 0.20, \qquad \tau_c = 0.25, \qquad \tau_d = 0.25, \tag{9}$$

where τ_α is the minimum rendering opacity, τ_c is the maximum color discrepancy, and τ_d is the maximum depth discrepancy. The reliable overlap region must contain at least 128 valid pixels.

When the strict inter-submap overlap mask contains fewer than 128 valid pixels, a relaxed visibility mask is used based on the source opacity, target opacity, and valid observed depth. The corresponding update is skipped if fewer than 128 valid pixels remain after this fallback.

Residual view-level registration. Although online pose registration provides the initial relationship between overlapping submaps, small residual view-level misalignments may remain. We therefore optimize an incremental relative transformation while keeping both Gaussian maps fixed. The residual pose is optimized for 30 iterations using Adam with learning rate

$$\eta_{\text{pose}} = 5 \times 10^{-3}. \tag{10}$$

The registration objective is

$$\mathcal{L}_{\text{pose}} = \mathcal{L}_c^{\text{obs}} + 0.5\,\mathcal{L}_d^{\text{obs}} + 0.25\,\mathcal{L}_c^{T} + 0.125\,\mathcal{L}_d^{T}, \tag{11}$$

where $\mathcal{L}_c^{\text{obs}}$ and $\mathcal{L}_d^{\text{obs}}$ denote color and depth consistency with the source observation, and $\mathcal{L}_c^{T}$ and $\mathcal{L}_d^{T}$ denote the corresponding overlap supervision from the target reference. After convergence, the target reference is rendered again using the refined relative transformation and the reliable overlap region is recomputed. The residual transformation is used only for the current inter-submap overlap refinement and is not written back to the global pose graph.

Gaussian refinement. After residual registration, the target submap and the refined relative transformation are fixed, while the source Gaussian representation is optimized for 60 iterations. The three principal loss weights are

$$(\lambda_{\text{local}}, \lambda_{\text{obs}}, \lambda_{\text{cross}}) = (0.50, 0.75, 0.25). \tag{12}$$

Here, $\mathcal{L}_{\text{local}}$ preserves the locally stable source representation, $\mathcal{L}_{\text{obs}}$ maintains consistency with the source agent's own RGB-D observations, and $\mathcal{L}_{\text{cross}}$ enforces agreement with the target reference in the reliable overlap region. An isotropic Gaussian-scale regularization term is additionally applied with unit weight.

After refinement, the updated source submap replaces its previous representation and remains available for subsequent tracking and submap registration. The refined representation can therefore influence subsequent inter-submap registration, completing the online reinforcement loop between pose registration and inter-submap overlap refinement.

A.4 Overlap-Aware Global Refinement Details

Before global refinement, the merged Gaussian representation is cleaned by removing low-confidence and redundant Gaussian primitives. The resulting map is then optimized using three successive stages.

Overlap-Aware Refinement. The first stage is optimized for 800 iterations. During overlap detection, at most 8000 Gaussians are sampled from each submap, and at most four matched view pairs are retained for each overlapping submap pair. At each iteration, one overlapping pair and one of its matched view pairs are sampled for optimization.

The same consistency thresholds as the inter-submap overlap refinement are used: $\tau_\alpha = 0.20$, $\tau_c = 0.25$, and $\tau_d = 0.25$. The reconstruction, inter-submap overlap color, and inter-submap overlap depth terms use unit weights, while each individual target-view contribution is weighted by 0.5.

Boundary-Focused Refinement. The second stage is optimized for 800 iterations. Submap starting frames, ending frames, and frames within ± 10 frames of overlap boundaries are treated as boundary candidates. During optimization, a boundary candidate is sampled with probability 0.8, while a keyframe from the remaining training views is sampled with probability 0.2. This sampling strategy allocates more updates to submap transitions, where geometric and appearance discontinuities are more likely to remain after fusion.

Global Refinement. The final stage performs 800 additional iterations using uniformly sampled training keyframes. This stage stabilizes the global appearance and geometry after the overlap-aware and boundary-focused updates.