MAGiSt3R: Multi-Agent Feed-forward 3D Reconstruction from Monocular RGB Videos

Ziren Gong[1] Xiaohan Li[2] Fabio Tosi[1] Ninghui Xu[3]
Stefano Mattoccia[1] Jianfei Cai[4] Matteo Poggi[1]

[1] University of Bologna, Italy
[2] Faculty of Dentistry, The University of Hong Kong, China
[3] School of Instrument Science and Engineering, Southeast University, China
[4] Monash University, Australia

Project page: https://zorangong.github.io/magist3r_page/

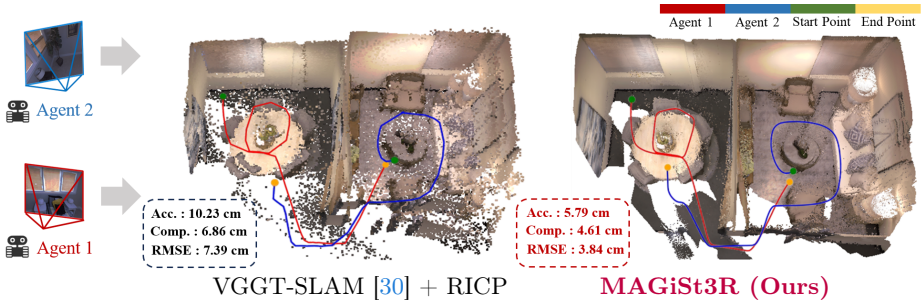

Fig. 1: MAGiSt3R – multi-agent 3D reconstruction in action. We showcase the mapping and tracking results of our method against VGGT-SLAM [30]+RICP (RANSAC+ICP) on ReplicaMultiagent *Apart 2*. Our framework produces more accurate and consistent reconstruction, along with precise camera trajectories.

Abstract. This paper presents MAGiSt3R, a multi-agent 3D reconstruction framework performing reconstruction and camera tracking for monocular RGB videos at almost 10 FPS. MAGiSt3R relies on a feed-forward model from the 3R family to process RGB videos and regress local point maps, and on a merging model, MAGMA, that combines local maps at both intra-agent and inter-agent levels to obtain the final, global point map. Furthermore, MAGiSt3R performs pose graph optimization to mitigate cumulative camera drift occurring along the feed-forward pipeline. We evaluate MAGiSt3R on both synthetic and real-world datasets, demonstrating its superior reconstruction and camera tracking accuracy compared to state-of-the-art feed-forward approaches.

1 Introduction

Multi-agent systems have emerged as a powerful paradigm in the broad field of artificial intelligence, enabling multiple autonomous entities to collaborate to achieve complex goals. Recent advances in agentic AI have further emphasized the importance of systems that, in addition to perceiving and reasoning about

the observed environment independently, can also proactively coordinate, share information, and act toward common objectives. This paradigm has also reached the intersection of robotics and computer vision, with the 3D reconstruction task, at the core of autonomous navigation and other higher-level applications, being lifted into a collaborative process, with multiple agents navigating independently in the environment and contributing to the mapping task [10, 17, 52].

Through the decades, 3D reconstruction has been approached in different flavors, mainly categorized into two families: offline approaches, such as Structure-from-Motion (SfM) [27, 28, 38, 39] and Multi-View Stereo (MVS), assuming the full availability of the images to process in advance; and online methods, such as Simultaneous Localization and Mapping (SLAM) systems [3, 11, 42], assuming image streams as the input. The latter is the one most suitable for real systems where, as soon as new images are collected, a prompt reaction is needed.

SLAM paradigm has faced a revolution [43] recently. Indeed, the adoption of Neural Radiance Fields (NeRF [32]) at first and 3D Gaussian Splatting (3DGS [22]) later as main engines for the mapping process has empowered SLAM systems to yield dense 3D reconstructions and possibly render the scene from novel, arbitrary viewpoints. This, however, came with significant memory requirements and runtime constraints, due to the need for performing per-scene optimization at deployment. On top of these advances, multi-agent SLAM systems [10, 52] have opened up new possibilities in spatial AI applications, possibly accelerating the reconstruction process while enforcing geometric consistency on the collaboratively reconstructed global map. Specifically, multi-agent systems built on top of NeRF [10] and 3DGS [52] engines have been proposed, respectively, yet inheriting the main limitations of NeRF and 3DGS, thus being i) scene-specific, since requiring the optimization process to take place directly during navigation, and ii) often unable to process under real-time constraints due to their iterative parameter updates.

In contrast, the advent of 3D vision foundation models, often referred to "3R" models [48], has ignited a new interest in developing feed-forward 3D reconstruction systems [12, 29, 30], complementing the latest NeRF/3DGS-based SLAM pipelines. On the one hand, these models can run on any unconstrained RGB video, not requiring any additional information such as depth – even with unknown camera parameters – and are no longer optimized over single sequences, thus allowing for much faster inference. On the other hand, 3R models processing video streams often lack global optimization, thus being prone to trajectory drift. In light of this, deploying 3R models in a multi-agent framework has the potential to exploit their strengths at best, while overcoming their main weaknesses related to camera drifting over long video sequences. However, no multi-agent feed-forward 3D reconstruction system has been developed to date.

Driven by these arguments, we introduce MAGiSt3R, the first multi-agent feed-forward 3D reconstruction framework for monocular RGB videos. Built on the latest advances in 3R models, MAGiSt3R globally and incrementally reconstructs scenes where multiple agents navigate concurrently. This is achieved according to the following three main steps: i) **Intra-agent reconstruction**: each

agent processes the input RGB sequence in sub-maps through a 3R backbone, then merges the sub-maps incrementally into a geometrically consistent one; ii) **Inter-agent alignment**: MAGiSt3R leverages a loop closure mechanism to detect overlapping regions between multiple agents. Once a loop is detected, the point maps from multiple agents are combined into the final global map; iii) **Pose graph optimization**: tracked poses are optimized after merging sub-maps at the intra-agent or inter-agent level, mitigating the influence of cumulative drift from previous steps. At the intersection of the three, a novel feed-forward model, namely Multi-Agent Global Map Aggregation (**MAGMA**), is responsible for merging submaps both at the intra-agent and inter-agent levels.

We demonstrate through extensive experiments that MAGiSt3R achieves higher-quality scene reconstruction, as shown in Figure 1, along with accurate pose estimation at almost 10 FPS. Our contributions can be summarized as follows:

- We propose the first multi-agent, feed-forward dense scene reconstruction framework for RGB videos, that incrementally reconstructs 3D point maps in a unified coordinate system without known camera parameters.
- We propose a custom Multi-Agent Global Map Aggregation module suitable for sub-map merging at both intra-agent and inter-agent levels, producing global point maps and accurate camera poses.
- We lift several existing, feed-forward 3D reconstruction models to the multi-agent setting. Compared with them, MAGiSt3R achieves state-of-the-art results in both 3D reconstruction and camera tracking.

2 Related Work

We briefly review the literature concerning 3D reconstruction, classifying existing methods into offline, SLAM, 3R-based systems and multi-agent SLAM.

Offline 3D Reconstruction. Methods belonging to this category assume the availability of a complete set of images beforehand [27, 28, 38, 39]. If images alone are available, both camera parameters and 3D points can be estimated via Structure from Motion (SfM) [1] algorithms. On the contrary, if cameras are known in advance, 3D reconstruction is carried out through Multi-View Stereo (MVS) [44]. More modern approaches have been proposed for both settings, ranging from implementing NeRF [32] or 3DGS [22]-like frameworks for 3D surface reconstruction [4–6, 15, 26] in the latter, or training large foundation [47] models in the former. Offline processing, however, is often not practical for real systems where agents navigate and interact with the environment.

Online SLAM systems. Traditional SLAM systems like ORB-SLAM [3] rely on sparse feature matching, while deep learning approaches such as DROID-SLAM [42] use learned features and dense bundle adjustment. However, these methods only produce sparse or semi-dense maps, lacking the capabilities for photorealistic reconstruction. Neural implicit representations have recently revolutionized dense SLAM systems [13, 14, 43]. NeRF-based approaches like iMap

[40] pioneered the use of MLPs for joint mapping and tracking, while NICE-SLAM [56] introduced hierarchical feature grids to improve scalability. Subsequent works like Vox-Fusion [50], Point-SLAM [36], and Co-SLAM [46] further enhance reconstruction quality through adaptive voxel allocation, neural point clouds, and hybrid coordinate encodings. GO-SLAM [53] integrates loop closing and bundle adjustment for global optimization and improved consistency. On the other hand, 3DGS has emerged as an alternative explicit representation in this field. GS-SLAM [49], Photo-SLAM [18], SplaTAM [21], and MonoGS [31] demonstrate real-time rendering capabilities while maintaining high photorealistic quality. Further works improve upon these methods to enhance tracking accuracy [16,25,55], mapping efficiency [34], and leverage multi-modal inputs [41]. Despite these advances, radiance-field representations remain frame-to-model approaches, mostly operating in the RGB-D setting, that require iterative parameter updates and per-scene optimization, limiting their speed in real-time applications.

Online 3R Models. Recent foundation models bypass scene-specific training to offer direct 3D reconstruction [9]. This began with DUSt3R [48] predicting point maps from image pairs in an end-to-end manner, which MASt3R [24] improved via dense feature matching. More recently, VGGT [47] advanced the field by jointly predicting point clouds, poses, and intrinsics from arbitrary image sequences in a single forward pass. Building on these works, novel SLAM systems have emerged. MASt3R-SLAM [33] leverages MASt3R for real-time monocular SLAM without requiring camera calibration, employing efficient point map matching and Sim(3) optimization. SLAM3R [29] introduces an end-to-end system with Image-to-Points (I2P) and Local-to-World (L2W) modules for progressive alignment. VGGT-SLAM [30] adopts the more powerful VGGT transformer, optimizing over the SL(4) manifold to handle projective ambiguities in uncalibrated settings. Spann3R [45], instead, uses spatial memory to maintain global consistency during incremental reconstruction, SLAM3R [29] implements a local to global alignment module, while Ov3R [12] tackles open-vocabulary 3D semantic reconstruction. While these methods are faster than NeRF/3DGS-based SLAM systems and could be optimal candidates to operate in multi-agent settings, existing frameworks operate in single-agent settings and lack robust mechanisms for multi-agent collaboration and global map consistency.

Multi-Agent SLAM. Multi-agent SLAM can be categorized into *centralized* and *distributed* architectures. Traditional centralized systems like CCM-SLAM [37] and CVI-SLAM [20] use a central server for map fusion and global optimization, while distributed approaches such as Swarm-SLAM [23] support peer-to-peer communication with robust inter-agent loop closure. Recent multi-agent frameworks like CP-SLAM [17] (distributed-to-centralized) and MNE-SLAM [10] (peer-to-peer) have integrated neural representations; however, both suffer from the computational overhead inherent to implicit neural methods. MAGiC-SLAM [52] leverages 3DGS for faster rendering and supports multiple agents through centralized coordination, achieving improved tracking via loop closure mechanisms. However, these methods are inherently RGB-D SLAM systems that re-

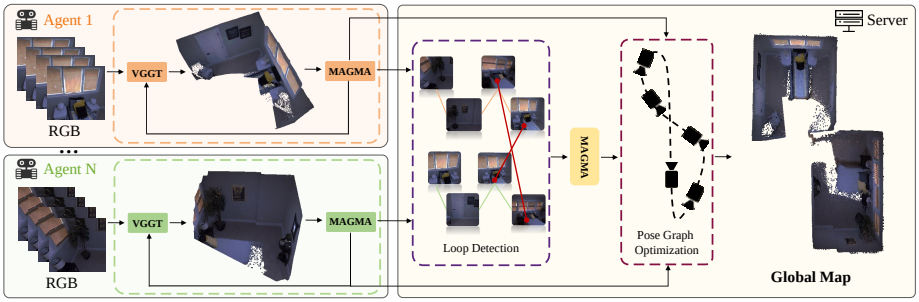

Fig. 2: Overview of MAGiSt3R. Given RGB sequences from multiple agents, our method simultaneously predicts local point maps and camera poses. Then, the MAGMA model merges submaps at the intra-agent level. On the server side, when loops are detected between agents, the same MAGMA module fuses the local maps of multiple agents into a global map. Finally, pose graph optimization further mitigates cumulative drift.

quire known camera parameters, perform slow scene-specific optimization, and focus on view synthesis rather than 3D reconstruction. In contrast, our MAGiSt3R builds upon feed-forward 3R models to achieve multi-agent collaboration without neural representation optimization for the first time. This enables the handling of unknown camera parameters via direct point map prediction at almost 10 FPS, extending applicability to RGB-only online 3D reconstruction.

3 Method

We now introduce MAGiSt3R, whose architecture is shown in Figure 2.

3.1 Intra-agent Reconstruction

We begin by describing how each agent independently processes its RGB sequence to generate local submaps.

Incremental Submap Generation. Following 3R-based systems, each agent in MAGiSt3R incrementally reconstructs scenes in its own unified coordinate system by predicting multiple submaps, e.g., n. Specifically, for any new set $\boldsymbol{I}_i = \{I_1^i, ..., I_m^i\}$ of m consecutive RGB images, a submap $\boldsymbol{S}_i$ is created, while the middle frame $I_{\frac{m}{2}}^i$ is labeled as a keyframe and added to a set of keyframes $\boldsymbol{I}_{key} = \{I_{\frac{m}{2}}^1, ..., I_{\frac{m}{2}}^n\}$, used later to support submap merging into a single one. Purposely, we use a 3R model, VGGT [47], to encode $\boldsymbol{I}_i$ into camera tokens $\boldsymbol{t}_i = \{t_1^i, ..., t_m^i\}$, and then predict dense depth maps $\boldsymbol{D}_i = \{D_1^i, ..., D_m^i\}$, confidence score maps $\boldsymbol{C}_i = \{C_1^i, ..., C_m^i\}$, and camera intrinsic and extrinsic parameters $\boldsymbol{P}_i = \{P_1^i, ..., P_m^i\}$:

$$\boldsymbol{t}_i, \boldsymbol{D}_i, \boldsymbol{C}_i, \boldsymbol{P}_i = \Phi_{\text{VGGT}}(\boldsymbol{I}_i) \tag{1}$$

Following [47], we obtain point maps $\boldsymbol{X}_i$ for frames in $\boldsymbol{I}_i$ by inverse projecting depth maps $\boldsymbol{D}_i$ according to camera parameters $\boldsymbol{P}_i$, in the reference system of

the first camera. We then filter $\boldsymbol{X}_i$ according to $\boldsymbol{C}_i$, removing any 3D points with confidence lower than τ_{conf}.
Concurrently, for frames in $\boldsymbol{I}_i$ we extract visual tokens $\boldsymbol{v}_i = \{v_1^i, ..., v_m^i\}$ with a ViT encoder E_{ViT}, as well as spatial tokens $\boldsymbol{\vartheta}_i = \{\vartheta_1^i, ..., \vartheta_m^i\}$ and global descriptors $\boldsymbol{\delta}_i = \{\delta_1^i, ..., \delta_m^i\}$ using DINOv2 SALAD [19]:

$$\boldsymbol{v}_i = E_{\text{ViT}}(\boldsymbol{I}_i) \qquad \boldsymbol{\delta}_i, \boldsymbol{\vartheta}_i = E_{\text{SALAD}}(\boldsymbol{I}_i) \tag{2}$$

These tokens are crucial for properly detecting loops and merging submaps into the final map, as we will detail in the remainder. We therefore define single submaps as $\boldsymbol{S}_i = \{\boldsymbol{t}_i, \boldsymbol{X}_i, \boldsymbol{P}_i, \boldsymbol{v}_i, \boldsymbol{\delta}_i, \boldsymbol{\vartheta}_i\}$ and, concurrently, define $\boldsymbol{S}_{key} = \{\boldsymbol{t}_{key}, \boldsymbol{X}_{key}, \boldsymbol{P}_{key}, \boldsymbol{v}_{key}, \boldsymbol{\delta}_{key}, \boldsymbol{\vartheta}_{key}\}$ by retaining the predictions obtained from frames in $\boldsymbol{I}_{key}$. This latter set will be instrumental in performing submap merging and loop closure.
According to this scheme, a single agent extracts n submaps along the scene. However, these have different reference systems – i.e., each one in the reference system of its first frame I_1^i. To merge these submaps into a single map in a shared reference system, we introduce our MAGMA model.

3.2 MAGMA Model

Fig. 3 illustrates MAGMA (Multi-Agent Global Map Aggregation), which incrementally aligns new submaps into a global coordinate system. It processes a reference set $\boldsymbol{S}_{ref}$ and a registering set $\boldsymbol{S}_{reg}$, representing the global map and the newly predicted submap demanding merging, respectively. These are forwarded to a geometric encoder E_{geo} and two decoders: a registering decoder D_{reg} that predicts point maps and poses aligned to the reference set, and a reference decoder D_{ref} that refines the original reference set.
Reference Set Retrieval. Inputs $\boldsymbol{S}_{reg}$ and $\boldsymbol{S}_{ref}$ to MAGMA are retrieved respectively from a local submap $\boldsymbol{S}_l$ demanding merging, and a global set $\{\boldsymbol{S}_g, \boldsymbol{S}_{key}\}$. After the single agent has processed images in $\boldsymbol{I}_i$, the local submap $\boldsymbol{S}_l$ to merge is $\boldsymbol{S}_i$, the global map $\boldsymbol{S}_g$ contains previously merged submaps $\{\boldsymbol{S}_{i-1}, \boldsymbol{S}_{i-2}, ...\}$, while the keyframe set $\boldsymbol{S}_{key}$ provides historic observations.
To retain only the most relevant views, we select the top-N best-correlated from $\{\boldsymbol{S}_g, \boldsymbol{S}_{key}\}$ as the reference set for submap merging. Specifically, we measure the visual similarity between descriptors in $\boldsymbol{\delta}_l$ and those in $\boldsymbol{\delta}_g$ and $\boldsymbol{\delta}_{key}$ to obtain correlation scores. Based on these scores, we select the most relevant N views from $\{\boldsymbol{S}_g, \boldsymbol{S}_{key}\}$ for $\boldsymbol{S}_{ref}$, while we retain $\boldsymbol{S}_l$ as the registering set $\boldsymbol{S}_{reg}$.
MAGMA Encoder. MAGMA encodes reference and registering point maps $\boldsymbol{X}_{ref}, \boldsymbol{X}_{reg}$ into geometry tokens $\boldsymbol{\gamma}_{ref} = \{\gamma_1, ..., \gamma_N\}_{ref}$ and $\boldsymbol{\gamma}_{reg} = \{\gamma_1, ..., \gamma_m\}_{reg}$

$$\boldsymbol{\gamma}_{ref} = E_{geo}(\boldsymbol{X}_{ref}), \quad \boldsymbol{\gamma}_{reg} = E_{geo}(\boldsymbol{X}_{reg}) \tag{3}$$

E_{geo} includes a 2D conv layer with kernel size 16 and stride 16, similar to the ViT patch embedding process, making the geometry tokens γ_i spatially aligned with visual tokens $\boldsymbol{v}_i$ to capture structural geometric primitives. However, geometry

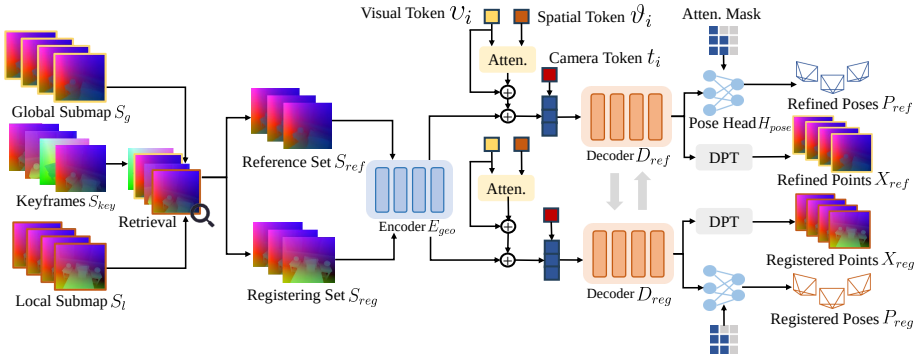

Fig. 3: Overview of MAGMA model. Given a global submap, a keyframe set, and a local submap, the MAGMA module merges the local submap into a global map with poses in a unified coordinate system. First, we use retrieval to obtain the most relevant frames to form the inputs, reference set and registering set. Next, we encode their point maps and aggregate these geometric features with visual tokens, spatial tokens, and camera tokens. Then, we decode and predict global point maps.

features alone may be insufficient for merging due to repetitive structures and lack of texture in point maps.

To address this issue, we introduce appearance tokens $\boldsymbol{\alpha}_{ref} = \{\alpha_1, ..., \alpha_N\}_{ref}$ for the reference set to enhance the geometry tokens and establish correspondences across different coordinate systems. These are derived from the previously introduced visual and spatial tokens through an attention module:

$$\boldsymbol{\alpha}_{ref} = \boldsymbol{v}_{ref} + softmax(\frac{\boldsymbol{v}_{ref}\boldsymbol{\vartheta}_{ref}^T}{\sqrt{d}})\boldsymbol{\vartheta}_{ref} \tag{4}$$

where d denotes the feature dimension. We apply the same process to obtain $\boldsymbol{\alpha}_{reg}$. The appearance tokens establish correspondences across different coordinate systems and mitigate the ambiguity inherent in geometry tokens alone. Geometry and appearance tokens are then combined and concatenated with camera tokens into integrated features $\boldsymbol{F}_{ref} = \{\mathcal{F}_1, ..., \mathcal{F}_N\}_{ref}$:

$$\boldsymbol{F}_{ref} = [\boldsymbol{t}_{ref}; \boldsymbol{\alpha}_{ref} + \boldsymbol{\gamma}_{ref}] \tag{5}$$

We apply the same process to obtain $\boldsymbol{F}_{reg}$.

MAGMA Decoders. Registering and reference decoders D_{reg}, D_{ref} process $\boldsymbol{F}_{ref}, \boldsymbol{F}_{reg}$, respectively. Each decoder block contains self-attention and multi-view cross-attention layers, followed by an MLP. The registering decoder first applies self-attention with $\boldsymbol{F}_{reg}$, then performs cross-attention with $\boldsymbol{F}_{reg}$ as queries and $\boldsymbol{F}_{ref}$ as keys and values. Finally, after max-pooling, we obtain registering tokens $\boldsymbol{G}_{reg} = \{\mathcal{G}_1, ..., \mathcal{G}_m\}_{reg}$ and reference tokens $\boldsymbol{G}_{ref} = \{\mathcal{G}_1, ..., \mathcal{G}_N\}_{ref}$:

$$\boldsymbol{G}_{reg} = D_{reg}(\boldsymbol{F}_{reg}, \boldsymbol{F}_{ref}), \quad \boldsymbol{G}_{ref} = D_{ref}(\boldsymbol{F}_{ref}, \boldsymbol{F}_{reg}) \tag{6}$$

MAGMA Heads. Finally, DPT heads [35] H_{pts} regress the refined point map $\boldsymbol{X'}_{ref}$, the registered point map $\boldsymbol{X'}_{reg}$, and confidence maps $\boldsymbol{C}_{ref}, \boldsymbol{C}_{reg}$

$$\boldsymbol{X'}_{ref}, \boldsymbol{C}_{ref} = H_{pts}(\boldsymbol{G}_{ref}), \quad \boldsymbol{X'}_{reg}, \boldsymbol{C}_{reg} = H_{pts}(\boldsymbol{G}_{reg}) \tag{7}$$

By retrieving pose embeddings $\boldsymbol{t}_{ref}, \boldsymbol{t}_{reg}$ from $\boldsymbol{G}_{ref}, \boldsymbol{G}_{reg}$, we use a pose head H_{pose}, consisting of four self-attention layers followed by a linear layer, to predict the refined camera parameters $\boldsymbol{P}_{ref}$ and the registered camera parameters $\boldsymbol{P}_{reg}$, with the latter transformed into the global coordinate system. Specifically, we concatenate the pose embeddings as $(\boldsymbol{t}_{ref}, \boldsymbol{t}_{reg})$. Then, we use an attention mask in the pose head to guide the information exchange: $\boldsymbol{t}_{ref}$ attends to all its tokens, while $\boldsymbol{t}_{reg}$ attends only to tokens in $\boldsymbol{t}_{ref}$.

$$\boldsymbol{P}_{ref}, \boldsymbol{P}_{reg} = H_{pose}(\boldsymbol{t}_{ref}, \boldsymbol{t}_{reg}) \tag{8}$$

The global map is then updated in its point maps and poses accordingly.

Training Loss. We train MAGMA end-to-end by supervising submap fusion. Following DUSt3R [48], we compute a confidence-aware registration loss between predicted point maps $\boldsymbol{X}_g$ in the global coordinate system and the ground truth point map $\hat{\boldsymbol{X}}_g$, normalized by the average Euclidean distance of valid points:

$$\mathcal{L}_{reg} = M \cdot \left(\boldsymbol{C}_g \cdot \left\| \frac{\boldsymbol{X}_g}{z} - \frac{\hat{\boldsymbol{X}}_g}{\hat{z}} \right\| - \beta \log \boldsymbol{C}_g \right) \tag{9}$$

where M is the mask of valid points with ground truth, z and $\hat{z}$ are scale factors, and β is a hyperparameter for regularization. We also enforce a camera loss between camera parameters $\boldsymbol{P}_g$ and ground truth ones $\hat{\boldsymbol{P}}_g$ as

$$\mathcal{L}_{pose} = \left\| \boldsymbol{P}_g - \hat{\boldsymbol{P}}_g \right\| \tag{10}$$

where $\hat{\boldsymbol{P}}_g = [q, b, f]$ contains a quaternion $q \in \mathbb{R}^4$, the translation vector $b \in \mathbb{R}^3$, and the field of view $f \in \mathbb{R}^2$, assuming the principal point at the image center. Finally, to ensure that the registered submap is consistently aligned with the global scene, we add a geometry consistency term. By randomly selecting two views j, k in the predicted submaps, we use ground truth depth maps $\hat{D}_j, \hat{D}_k$ and intrinsics $\hat{K}_j, \hat{K}_k$, together with the predicted camera poses T_j, T_k derived from $\boldsymbol{P}_g$, to reproject pixels from view j to view k. We then measure the distance between projected point maps $\tilde{X}_j$ and $\tilde{X}_k$ in the global coordinate system as:

$$\mathcal{L}_{geo} = M \left\| \tilde{X}_j - \tilde{X}_k \right\|_2 \tag{11}$$

where M is the mask of valid pixels. The total training loss $\mathcal{L}$ is then defined as:

$$\mathcal{L} = \lambda_{reg}\mathcal{L}_{reg} + \lambda_{pose}\mathcal{L}_{pose} + \lambda_{geo}\mathcal{L}_{geo} \tag{12}$$

3.3 Inter-agent Reconstruction

We now present how the MAGMA model combines the efforts of the single agents into a global 3D reconstruction.

Loop Detection. To achieve global and consistent reconstruction across multiple agents, we need to detect regions where the trajectories of the different agents overlap. We set the first camera of the first agent A_1 as the global coordinate system. Then, when a new submap is created by another agent, e.g., A_2, we look for loops between this submap and keyframes $\boldsymbol{S}_{key}^{A_1}$ from A_1, by computing correlations between $\boldsymbol{\delta}_i^{A_2}$ and $\boldsymbol{\delta}_{key}^{A_1}$. When the maximum correlation score exceeds a threshold τ_{loop}, a loop between the agents is detected.

Inter-agent Submap Fusion. Once a loop is detected, we take the submap from the first agent A_1 containing the keyframe with the maximum correlation score as the global submap $\boldsymbol{S}_g^{A_1}$ and forward it to MAGMA, together with local submap $\boldsymbol{S}_l^{A_2}$ (the latest submap reconstructed by the other agent A_2) and the keyframe set $\boldsymbol{S}_{key}^{A_2}$. We retrieve the top-$N$ most-correlated views from the $\left\{\boldsymbol{S}_l^{A_2}, \boldsymbol{S}_{key}^{A_2}\right\}$ as the registering set $\boldsymbol{S}_{reg}$, where $\boldsymbol{S}_{key}^{A_2}$ serves as historic observations providing global clues. Concurrently, we retain $\boldsymbol{S}_g^{A_1}$ as the reference set $\boldsymbol{S}_{ref}$. Then, MAGMA predicts the registered point map and poses for agent A_2, as described in Sec. 3.2, which are further used to compute the relative transformation between local and global coordinates and update the global map.

3.4 Backend Pose Graph Optimization

We further refine predicted poses through classical optimization techniques [52].

Graph Construction. To mitigate cumulative errors during the incremental process, we perform pose graph optimization (PGO) after submap merging to reduce drift and enforce global consistency. Each submap created by an agent corresponds to a node $\boldsymbol{n}_i \in SE(3)$ in the graph. The edges e_{ij} between the neighboring nodes represent the relative transformations computed by the proposed MAGMA model during the intra-agent stage.

Loop Closure. We compute the visual similarity between the extracted global descriptors $\boldsymbol{\delta}_i$ of the keyframes in each agent to retrieve potential loop candidates for a new submap. If a correlation score higher than τ_{loop} is found, a loop is detected. The loop edge is estimated by MAGMA during the inter-agent stage.

Optimization. We obtain the optimized camera poses by leveraging the Levenberg-Marquardt algorithm to minimize the objective function:

$$\min \sum_{i,j \in \varepsilon} \left\| log(e_{ij}^{-1}(n_i^{-1} n_j)) \right\|_{\Omega_{ij}}^2 \tag{13}$$

where ε denotes the index of nodes. n_i and n_j are node poses, and log is the logarithmic map from $SE(3)$ to $\mathfrak{se}(3)$. Ω_{ij} is the information matrix reflecting the uncertainty over edges – the higher the score, the higher the weight.

Pose Update Integration. After performing pose graph optimization, we obtain the pose corrections $\boldsymbol{n}_i'$ for each submap of each agent. All camera poses $\boldsymbol{T}_i = \{T_1^i, ..., T_m^i\}$ in each submap are updated as:

$$\boldsymbol{T}_i = \boldsymbol{n}_i' \boldsymbol{T}_i \tag{14}$$

4 Experiments

We now present a thorough evaluation of MAGiSt3R, including implementation details, ablation studies, and comparisons with state-of-the-art methods.

Dataset. We train our Multi-Agent Global Map Aggregation (MAGMA) module on a mixture of several datasets: ScanNet [7], ScanNet++ [51], and Aria Synthetic Environment [2]. ScanNet and ScanNet++ provide real-world, diverse indoor environments, ranging from small rooms to large-scale indoor scenarios. Aria Synthetic Environment offers photorealistic multi-room scenes. To effectively learn how to merge point maps both at the intra-agent and inter-agent levels, MAGMA is trained by partitioning the diverse scenes in the training set to simulate a multi-agent system – since none of the training datasets support this setting natively. Specifically, two trajectories are created starting from the first and last frames of a scene, sampling one frame every 2 for each trajectory. For evaluation, we perform 3D reconstruction and camera tracking on the ReplicaMultiagent [17] and the AriaMultiagent [52] datasets. The former offers synthetic single and multiple rooms where 2 agents navigate independently and is specifically tailored for 3D reconstruction, while the latter provides real-world indoor environments explored by 3 agents, with ground truth camera information and depth available for evaluation, yet no ground truth reconstructions.

Implementation Details. We implement MAGiSt3R in PyTorch. Specifically, we train MAGMA for 200 epochs with batch size 5 on 4 A100 GPUs and learning rate $1.5e^{-5}$. We set the number of input images processed by VGGT $m = 10$ with stride $s = 5$, the confidence threshold $\tau_{conf} = 0.25$, the correlation threshold $\tau_{loop} = 0.6$, top-correlated samples $N = 10$, the regularization term $\beta = 1$, and the loss weights $\lambda_{reg} = 1, \lambda_{pose} = 1, \lambda_{geo} = 0.8$. Following the literature on multi-agent SLAM [10, 52], we run our evaluation in a multi-agent setting, while also reporting the results achieved by the single agents running independently.

Baselines. For single-agent evaluation, we compare our MAGiSt3R against state-of-the-art feed-forward frameworks: DUSt3R [48] and MASt3R [24] processing two views at a time sequentially, SLAM3R [29], MASt3R-SLAM [33], and VGGT-SLAM [30] as incremental multi-view 3D reconstruction methods. For the multi-agent setting, existing neural SLAM systems [10, 17, 52] require RGB-D input; therefore, we report them as references rather than fair baselines when evaluating tracking accuracy. Instead, we adapt [24, 29, 30, 48] for multi-agent operation by independently processing each agent's video and merging the resulting point maps using RANSAC + ICP alignment (RICP), providing comparable feed-forward baselines, and we compare against a concurrent work lifting MASt3R-SLAM [33] to multi-agent setting through global graph optimization – referred to as MA-MASt3R-SLAM [54].

Metrics. For quantitative evaluation, we follow [29] and build ground truth point clouds by projecting pixels to 3D points using ground truth depth and camera parameters. Reconstruction quality is evaluated through accuracy and completeness, tracking accuracy using Absolute Trajectory Error (ATE) RMSE.

4.1 Ablation Studies

We start our evaluation by running ablation experiments on ReplicaMultiagent.

Table 1: Ablation study – Impact of inter-agent merging strategies. Results in the multi-agent setting.

Methods	Train Setting	Acc.	Comp.	RMSE
(A) w/ RICP	×	8.89	6.47	7.66
(B) w/ SL(4)	×	13.66	11.67	18.94
(C) w/ L2W	Multi Agent	9.71	7.16	9.19
(D) w/ MAGMA	Single Agent	6.89	4.93	5.35
(E) **MAGiSt3R**	Multi Agent	**5.37**	**3.36**	**3.87**
(F) w/o L_{geo}	Multi Agent	5.88	3.60	3.97
(G) w/o Spatial Tokens	Multi Agent	6.04	3.71	4.19

Impact of MAGMA. Tab. 1 demonstrates the effectiveness of MAGMA at combining the submaps from different agents, by comparing it with different merging strategies on the Replica-Multiagent dataset. In this experiment, we report both reconstruction and camera tracking metrics in the multi-agent setting. We select three typical fusion strategies as baselines, including (A) RANSAC+ICP (RICP) used by MAGiC-SLAM [52], (B) optimization over SL(4) manifold used by VGGT-SLAM [30], and (C) the L2W module proposed in SLAM3R [29]. Finally, (D) we also evaluate the effectiveness of MAGMA in the multi-agent setting, yet trained in a single-agent regime only. All variants use VGGT as the single-agent backbone. We can notice how MAGMA consistently outperforms existing alternatives, even when not trained to deal with multiple agents (D) – while a proper training under multi-agent assumptions unleashes its full potential (E). We observe that both RICP (A) and SL(4) (B) heavily rely on the similarity of two fused submaps: the less similar the two subsets, the poorer the alignment obtained when merging. This makes them well-suited for incremental intra-agent alignment, yet ineffective for operating at the inter-agent level. In contrast, our MAGMA module learns corresponding matches using geometry features across the subsets, remaining effective even with lower overlaps. The L2W branch from SLAM3R [29] also underperforms at this task, due to the poorer geometry and visual information it processes. On a side note, RICP achieves the best results among the other methods, motivating our choice of using it to build the multi-agent feed-forward baselines for the rest of our experiments. Finally, we report the results achieved by ablated versions of MAGMA in (F) and (G), respectively, by removing L_{geo} loss and the spatial tokens, confirming that both play an important role.

Effects of Backbone. We compare using VGGT against an alternative backbone, the recent state-of-the-art model SAIL-Recon [8]. This experiment is reported in Tab. 2 row (H). The results shown by our original framework (E) demonstrate that selecting VGGT yields better results.

Table 2: Ablation study – Impact of backbones and PGO. Results in the multi-agent setting.

Methods	Acc.	Comp.	RMSE
(E) **MAGiSt3R**	**5.37**	**3.36**	**3.87**
(H) w/ SAIL-Recon [8]	5.97	3.88	4.06
(I) w/o PGO	6.18	4.11	5.65

Table 3: Reconstruction results on ReplicaMultiagent. $^{\text{II}}$ means using RICP at the inter-agent level. The best results are shown as **first**, second, and third.

Methods	Setting	Apart 0		Apart 1		Apart 2		Office 0		Avg.	
		Acc.	Comp.	Acc.	Comp.	Acc.	Comp.	Acc.	Comp.	Acc.	Comp.
DUSt3R [48]	Agent 1	6.95	3.85	15.78	12.29	10.15	9.17	12.63	12.43	11.38	9.44
MASt3R [24]		5.01	3.00	9.51	3.98	7.32	3.91	6.14	5.30	7.00	4.05
SLAM3R [29]		5.17	4.03	9.18	6.91	4.97	4.94	10.18	7.10	7.38	5.75
MASt3R-SLAM [33]		10.91	6.35	10.38	3.24	9.68	**3.28**	7.58	6.91	9.64	4.94
VGGT-SLAM [30]		4.59	2.89	15.64	9.66	9.87	4.15	6.06	4.80	9.04	5.38
MAGiSt3R (ours)		**4.07**	**2.56**	**5.57**	**3.05**	**3.96**	3.35	**5.01**	**2.95**	**4.65**	**2.98**
DUSt3R [48]	Agent 2	6.87	4.69	14.14	11.82	7.88	**5.02**	6.98	6.26	8.97	6.95
MASt3R [24]		4.95	2.48	15.78	12.87	10.87	12.39	7.66	7.68	9.82	8.86
SLAM3R [29]		7.24	7.68	15.18	14.80	8.73	8.73	5.98	**4.36**	9.28	8.89
MASt3R-SLAM [33]		13.09	4.66	21.57	23.93	13.19	14.91	8.06	8.31	13.98	12.95
VGGT-SLAM [30]		4.29	2.37	14.41	11.95	6.99	6.15	7.45	6.78	8.29	6.81
MAGiSt3R (ours)		**3.43**	**1.79**	**9.26**	**8.94**	**6.89**	5.72	**5.26**	4.57	**6.21**	**5.26**
DUSt3R [48] $^{\text{II}}$	Multi-Agent	11.79	8.72	15.21	11.71	11.24	10.24	11.80	9.03	12.51	9.92
MASt3R [24] $^{\text{II}}$		6.29	4.83	12.97	9.31	12.03	10.48	9.18	8.01	10.12	8.16
SLAM3R [29] $^{\text{II}}$		18.09	10.79	16.57	15.25	14.38	6.44	8.11	3.75	14.29	9.06
MASt3R-SLAM [33] $^{\text{II}}$		15.76	6.52	16.26	10.81	12.69	10.97	10.59	9.16	13.83	9.37
VGGT-SLAM [30] $^{\text{II}}$		8.60	5.92	15.97	13.00	10.23	6.86	10.65	7.97	11.36	8.44
MAGiSt3R (ours)		**4.20**	**2.57**	**8.32**	**4.26**	**5.79**	**4.61**	**3.18**	**2.01**	**5.37**	**3.36**

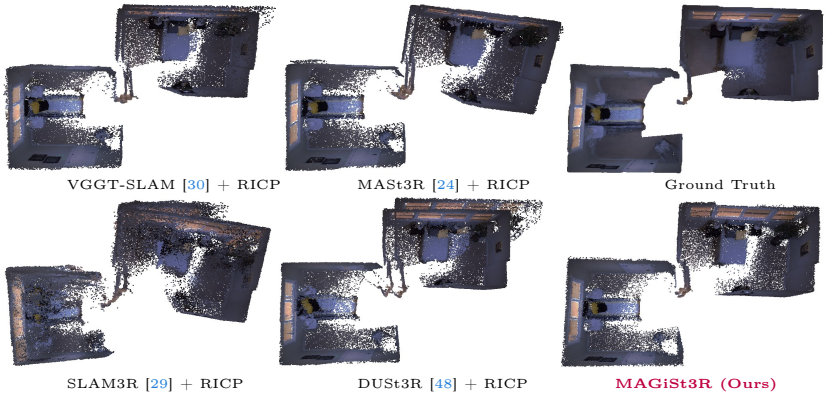

Fig. 4: Qualitative results – Dense point maps on ReplicaMultiagent. We present the reconstruction results with the multi-agent setting on the *Apart 0* sequence. Our method yields a more accurate and consistent global point map.

Effects of PGO. Finally, we measure the impact of Pose Graph Optimization (PGO) on our SLAM system. As shown in Tab. 2 row (I), removing this component yields significant drops in accuracy, making it more impactful than backbone selection. However, by comparing (I) with entries (A), (B), and (C) in Tab. 1, we can appreciate how MAGMA still outperforms existing merging strategies even without PGO.

4.2 Comparison with the State-of-the-Art

We now compare MAGiSt3R with existing feed-forward models.

3D Reconstruction on ReplicaMultiagent. In Tab. 3, we compare the 3D reconstruction quality of our method against the state-of-the-art feed-forward approaches on the ReplicaMultiagent dataset, containing both single-room and

Table 4: Tracking results on ReplicaMultiagent (single-agent).

Method	Agent 1					Agent 2				
	Apart 0	Apart 1	Apart 2	Office 0	Avg.	Apart 0	Apart 1	Apart 2	Office 0	Avg.
DUSt3R [48]	6.76	8.87	7.30	21.04	10.99	8.18	9.93	10.97	4.31	8.35
MASt3R [24]	**2.52**	4.46	**3.62**	9.56	5.04	3.58	**3.10**	3.45	**2.08**	3.05
SLAM3R [29]	7.25	30.62	8.77	18.43	16.27	14.12	27.94	11.35	18.65	18.02
MASt3R-SLAM [33]	4.19	**3.79**	6.04	10.57	6.15	7.90	9.29	5.02	5.40	6.90
VGGT-SLAM [30]	2.96	8.90	5.77	9.07	6.68	2.64	7.86	5.89	3.63	5.01
MAGiSt3R (ours)	2.54	6.99	3.88	**2.69**	**4.03**	**1.87**	4.06	**3.23**	2.48	**2.91**

multi-room environments. On top, we report the reconstruction accuracy and completion achieved by single agents independently – i.e., without any inter-agent communication – followed at the bottom by the results achieved under the multi-agent setting. It is worth highlighting that results achieved by single agents alone and by the multi-agent system as a whole are not directly comparable, as they cover different portions of the entire scene. By focusing on single agents, MAGiSt3R already outperforms any feed-forward model in both Agent 1 and Agent 2 tracks, with few exceptions – completeness on *Apart 2* and *Office 0*, supporting the superior capability of MAGMA to aggregate submaps at the intra-agent level. Furthermore, when moving to the multi-agent setting, our framework shines and largely outperforms all proposed baselines, confirming the effectiveness of our MAGMA model even at the inter-agent level, and therefore establishing it as a key component for both single and multiple agent systems. This fact is confirmed qualitatively by Fig. 4: reconstructions by MAGiSt3R expose significantly fewer artefacts with respect to the other methods that, on the contrary, struggle at properly aligning the two rooms explored in the *Apart 0* sequence, often reconstructing duplicated portions of the scene or with some parts being significantly drifted with respect to the global point map.

Tracking on ReplicaMultiagent. Tab. 4 reports tracking accuracy on ReplicaMultiagent, by the same methods when deployed in single-agent settings. Although MAGiSt3R achieves mixed results on individual scenes, our method still achieves the average best performance with both agents. Furthermore, Tab. 5 extends to the multi-agent setting, also reporting existing RGB-D SLAM systems designed for this specific setup [17,23,52] as upper bounds. In this case, MAGiSt3R consistently outperforms all the other multi-agent feed-forward baselines, confirming that MAGMA effectively improves inter-agent aggregation. On average, MAGiSt3R also outperforms the concurrent MA-MASt3R-SLAM [54], getting very close to RGB-D systems.

Table 5: Tracking results on ReplicaMultiagent (multi-agent). *means average excluding *Apart 2* to compare with [54].

Methods	Multi-Agent				
	Apart 0	Apart 1	Apart 2	Office 0	Avg.
RGB-D (Calibrated)					
Swarm-SLAM [23]	1.80	5.56	5.61	1.42	3.60
CP-SLAM [17]	0.95	1.42	1.91	0.65	1.23
MAGiC-SLAM [52]	0.16	0.26	0.32	0.27	0.25
RGB (Uncalibrated)					
DUSt3R [48] ‖	8.35	12.27	11.43	14.28	11.58
MASt3R [24] ‖	4.02	6.58	5.66	7.69	5.99
SLAM3R [29] ‖	11.72	31.92	11.74	20.40	18.94
MASt3R-SLAM [33] ‖	6.96	8.85	7.46	9.44	8.17
VGGT-SLAM [30] ‖	3.42	10.34	7.39	7.46	7.15
MA-MASt3R-SLAM [54]	5.24	**5.48**	-	**2.57**	- /4.43*
MAGiSt3R (ours)	**2.75**	5.81	**3.84**	3.09	**3.87/3.88***

Table 6: Tracking results on AriaMultiagent (single-agent).

Methods	Agent 1			Agent 2			Agent 3		
	Room 0	Room 1	Avg.	Room 0	Room 1	Avg.	Room 0	Room 1	Avg.
DUSt3R [48]	3.90	24.21	14.06	12.19	19.97	16.08	26.46	17.78	22.12
MASt3R [24]	2.95	29.12	16.04	8.47	16.12	12.30	8.82	14.69	11.76
SLAM3R [29]	4.16	10.17	7.17	7.10	3.72	5.41	6.65	3.36	5.01
MASt3R-SLAM [33]	3.92	15.31	9.62	9.28	47.65	28.47	9.87	15.22	12.55
VGGT-SLAM [30]	3.78	9.30	6.54	11.55	9.15	10.35	11.01	5.66	8.34
MAGiSt3R (ours)	**2.89**	**2.88**	**2.89**	**6.11**	**1.86**	**3.99**	**4.26**	**1.65**	**2.96**

AriaMultiagent. We extend our evaluation to AriaMultiagent, a real-world egocentric dataset supporting experiments for up to 3 agents. In Tab. 6, we report tracking accuracy achieved by feed-forward methods running independently in the single-agent settings. Unlike the ReplicaMultiagent experiments, in this case MAGiSt3R consistently outperforms other feed-forward methods on both *Room 0* and *Room 1* scenes, along any of the single agent trajectories.

In Tab. 7, we complete our evaluation by collecting results for the multi-agent setting. Once again, we report RGB-D systems at the top as a reference. We can appreciate how MAGiSt3R consistently outperforms any other multi-agent feed-forward method by a significant margin, highlighting, in particular, a notable gap with MA-MASt3R-SLAM [54]. This further confirms the crucial role played by MAGMA, outplaying merging strategies deployed by previous frameworks and concurrent multi-agent systems.

Table 7: Tracking results on AriaMultiagent (multi-agent).

Methods	Multi-Agent		
	Room 0	Room 1	Avg.
RGB-D (Calibrated)			
Swarm-SLAM [23]	6.45	4.78	5.62
CP-SLAM [17]	3.03	2.87	2.95
MAGiC-SLAM [52]	1.15	0.65	0.90
RGB (Uncalibrated)			
DUSt3R [48] ‖	15.46	21.36	18.41
MASt3R [24] ‖	7.80	20.68	14.24
SLAM3R [29] ‖	6.97	6.31	6.64
MASt3R-SLAM [33] ‖	9.13	29.34	19.24
VGGT-SLAM [30] ‖	9.59	8.47	9.03
MA-MASt3R-SLAM [54]	7.59	31.15	19.37
MAGiSt3R (ours)	**4.68**	**2.36**	**3.52**

Supplementary material. We refer the reader to the supplementary material for further experiments and insights.

4.3 Runtime Comparison with RGB-D Systems

Table 8: Runtime analysis. *taken from original paper.

Method	**FPS**
CP-SLAM [17]*	< 1
MAGiC-SLAM [52]*	2
DUSt3R [48] ‖	< 1
MASt3R [24] ‖	< 1
SLAM3R [29] ‖	3
MASt3R-SLAM [33] ‖	9
VGGT-SLAM [30] ‖	**23**
MA-MASt3R-SLAM* [54]	11
MAGiSt3R (ours)	9

In Tab. 8, we conduct a runtime analysis on ReplicaMultiagent in the two-agent setup. RGB-D SLAM systems achieve higher tracking accuracy, yet existing solutions achieve very few FPS. In contrast, VGGT-SLAM [30] shows the full speed potential of feed-forward models, although falling short on accuracy. Finally, both MA-MASt3R-SLAM [54] and MAGiSt3R run at ∼10 FPS, although our framework retrieves significantly more accurate trajectories on AriaMultiagent.

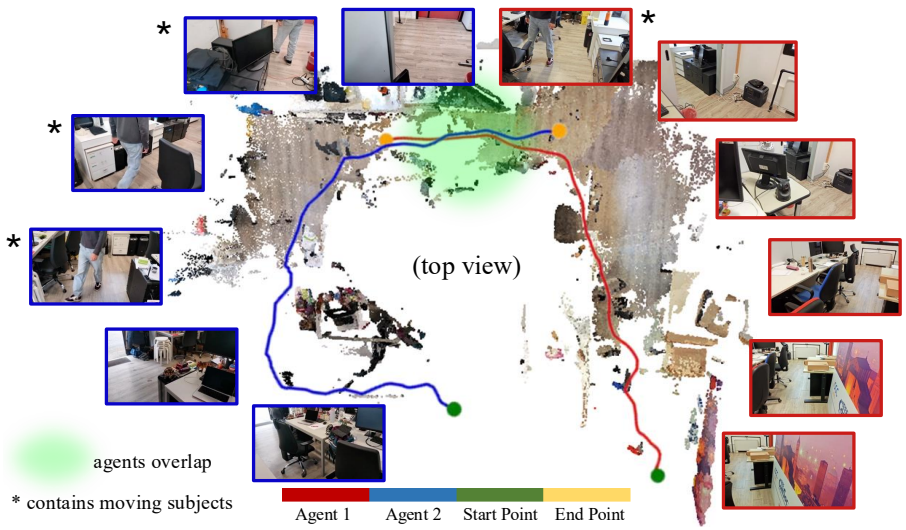

Fig. 5: Experiments on a self-collected scene with limited overlap. MAGiSt3R effectively works out globally consistent scene reconstruction.

5 Qualitative results on custom sequences

We conclude with qualitative results on real-world video sequences, collected by 2 agents in low overlap (∼15%) and in the presence of some subjects moving in the scene, thus representing a particularly challenging scenario that pushes the limits of standard multi-agent reconstruction pipelines. Fig. 5 shows the results by MAGiSt3R, looking qualitatively consistent despite such challenges.

6 Conclusion

In this paper, we presented the first multi-agent feed-forward RGB 3D reconstruction system, MAGiSt3R, achieving high-quality collaborative scene reconstruction and camera tracking in indoor scenes. Our feed-forward framework enables the system to reconstruct scenes at almost 10 FPS, with the proposed MAGMA model enabling MAGiSt3R to merge the submaps from both intra-agent and inter-agent. Our experiments demonstrate that MAGiSt3R achieves both accurate reconstruction and camera tracking.

Limitations and Future Work. Our current framework, as well as the baselines, is not meant for agents facing very different conditions – e.g., day vs night, adverse weather. We plan to extend this aspect, possibly with new benchmarks, as well as to evaluate multi-agent systems at a larger scale (> 3 agents).

Acknowledgment. The authors sincerely thank the scholarship supported by the China Scholarship Council (CSC). The authors also acknowledge the CINECA award under the ISCRA initiative for the availability of high-performance computing resources and support.

References

1. Arrigoni, F.: A taxonomy of structure from motion methods. arXiv preprint arXiv:2505.15814 (2025) 3
2. Avetisyan, A., Xie, C., Howard-Jenkins, H., Yang, T.Y., Aroudj, S., Patra, S., Zhang, F., Frost, D., Holland, L., Orme, C., et al.: Scenescript: Reconstructing scenes with an autoregressive structured language model. In: European Conference on Computer Vision. pp. 247–263. Springer (2024) 10
3. Campos, C., Elvira, R., Rodríguez, J.J.G., Montiel, J.M., Tardós, J.D.: Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam. IEEE transactions on robotics **37**(6), 1874–1890 (2021) 2, 3
4. Chen, D., Li, H., Ye, W., Wang, Y., Xie, W., Zhai, S., Wang, N., Liu, H., Bao, H., Zhang, G.: Pgsr: Planar-based gaussian splatting for efficient and high-fidelity surface reconstruction. arXiv preprint arXiv:2406.06521 (2024) 3
5. Chen, Y., Xu, H., Zheng, C., Zhuang, B., Pollefeys, M., Geiger, A., Cham, T.J., Cai, J.: Mvsplat: Efficient 3d gaussian splatting from sparse multi-view images. In: European Conference on Computer Vision. pp. 370–386. Springer (2024) 3
6. Chen, Y., Zheng, C., Xu, H., Zhuang, B., Vedaldi, A., Cham, T.J., Cai, J.: Mvsplat360: Feed-forward 360 scene synthesis from sparse views. Advances in Neural Information Processing Systems **37**, 107064–107086 (2024) 3
7. Dai, A., Chang, A.X., Savva, M., Halber, M., Funkhouser, T., Nießner, M.: Scannet: Richly-annotated 3d reconstructions of indoor scenes. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 5828–5839 (2017) 10
8. Deng, J., Li, H., Xie, T., Ren, W., Zhang, Q., Tan, P., Guo, X.: Sail-recon: Large sfm by augmenting scene regression with localization. arXiv preprint arXiv:2508.17972 (2025) 11
9. Deng, T., Pan, Y., Yuan, S., Li, D., Wang, C., Li, M., Chen, L., Xie, L., Wang, D., Wang, J., Civera, J., Wang, H., Chen, W.: What is the best 3d scene representation for robotics? from geometric to foundation models. arXiv preprint arXiv:2512.03422 (2025) 4
10. Deng, T., Shen, G., Xun, C., Yuan, S., Jin, T., Shen, H., Wang, Y., Wang, J., Wang, H., Wang, D., et al.: Mne-slam: Multi-agent neural slam for mobile robots. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 1485–1494 (2025) 2, 4, 10
11. Engel, J., Schöps, T., Cremers, D.: Lsd-slam: Large-scale direct monocular slam. In: European conference on computer vision. pp. 834–849. Springer (2014) 2
12. Gong, Z., Li, X., Tosi, F., Han, J., Mattoccia, S., Cai, J., Poggi, M.: Ov3r: Open-vocabulary semantic 3d reconstruction from rgb videos. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 34206–34216 (2026) 2, 4
13. Gong, Z., Li, X., Tosi, F., Zhang, Y., Mattoccia, S., Wu, J., Poggi, M.: Dino-slam: Dino-informed rgb-d slam for neural implicit and explicit representations. In: European Conference on Computer Vision (2026) 3
14. Gong, Z., Tosi, F., Zhang, Y., Mattoccia, S., Poggi, M.: Hs-slam: Hybrid representation with structural supervision for improved dense slam. In: 2025 IEEE International Conference on Robotics and Automation (ICRA). pp. 8464–8470. IEEE (2025) 3
15. Guédon, A., Lepetit, V.: Sugar: Surface-aligned gaussian splatting for efficient 3d mesh reconstruction and high-quality mesh rendering. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5354–5363 (2024) 3

16. Ha, S., Yeon, J., Yu, H.: Rgbd gs-icp slam. In: European Conference on Computer Vision. pp. 180–197. Springer (2024) 4
17. Hu, J., Mao, M., Bao, H., Zhang, G., Cui, Z.: Cp-slam: Collaborative neural point-based slam system. Advances in Neural Information Processing Systems **36**, 39429–39442 (2023) 2, 4, 10, 13, 14
18. Huang, H., Li, L., Cheng, H., Yeung, S.K.: 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 (2024) 4
19. Izquierdo, S., Civera, J.: Optimal transport aggregation for visual place recognition. In: Proceedings of the ieee/cvf conference on computer vision and pattern recognition. pp. 17658–17668 (2024) 6
20. Karrer, M., Schmuck, P., Chli, M.: Cvi-slam—collaborative visual-inertial slam. IEEE Robotics and Automation Letters **3**(4), 2762–2769 (2018) 4
21. Keetha, N., Karhade, J., Jatavallabhula, K.M., Yang, G., Scherer, S., Ramanan, D., Luiten, J.: 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) 4
22. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G.: 3d gaussian splatting for real-time radiance field rendering. ACM Transactions on Graphics **42**(4), 1–14 (2023) 2, 3
23. Lajoie, P.Y., Beltrame, G.: Swarm-slam: Sparse decentralized collaborative simultaneous localization and mapping framework for multi-robot systems. IEEE Robotics and Automation Letters **9**(1), 475–482 (2023) 4, 13, 14
24. Leroy, V., Cabon, Y., Revaud, J.: Grounding image matching in 3d with mast3r. In: European Conference on Computer Vision. pp. 71–91. Springer (2024) 4, 10, 12, 13, 14
25. Li, X., Gong, Z., Tosi, F., Poggi, M., Mattoccia, S., Liu, D., Wu, J.: Stereo 3d gaussian splatting slam for outdoor urban scenes. arXiv preprint arXiv:2507.23677 (2025) 4
26. Li, Z., Müller, T., Evans, A., Taylor, R.H., Unberath, M., Liu, M.Y., Lin, C.H.: Neuralangelo: High-fidelity neural surface reconstruction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8456–8465 (2023) 3
27. Lindenberger, P., Sarlin, P.E., Larsson, V., Pollefeys, M.: Pixel-perfect structure-from-motion with featuremetric refinement. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 5987–5997 (2021) 2, 3
28. Liu, S., Yu, Y., Pautrat, R., Pollefeys, M., Larsson, V.: 3d line mapping revisited. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 21445–21455 (2023) 2, 3
29. Liu, Y., Dong, S., Wang, S., Yin, Y., Yang, Y., Fan, Q., Chen, B.: Slam3r: Real-time dense scene reconstruction from monocular rgb videos. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 16651–16662 (2025) 2, 4, 10, 11, 12, 13, 14
30. Maggio, D., Lim, H., Carlone, L.: Vggt-slam: Dense rgb slam optimized on the sl (4) manifold. arXiv preprint arXiv:2505.12549 (2025) 1, 2, 4, 10, 11, 12, 13, 14
31. Matsuki, H., Murai, R., Kelly, P.H., Davison, A.J.: Gaussian splatting slam. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 18039–18048 (2024) 4

32. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. In: European Conference on Computer Vision. pp. 405–421. Springer (2020) 2, 3
33. Murai, R., Dexheimer, E., Davison, A.J.: Mast3r-slam: Real-time dense slam with 3d reconstruction priors. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 16695–16705 (2025) 4, 10, 12, 13, 14
34. Peng, Z., Shao, T., Liu, Y., Zhou, J., Yang, Y., Wang, J., Zhou, K.: Rtg-slam: Real-time 3d reconstruction at scale using gaussian splatting. In: ACM SIGGRAPH 2024 Conference Papers. pp. 1–11 (2024) 4
35. Ranftl, R., Bochkovskiy, A., Koltun, V.: Vision transformers for dense prediction. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 12179–12188 (2021) 8
36. Sandström, E., Li, Y., Van Gool, L., Oswald, M.R.: Point-slam: Dense neural point cloud-based slam. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 18433–18444 (2023) 4
37. Schmuck, P., Chli, M.: Ccm-slam: Robust and efficient centralized collaborative monocular simultaneous localization and mapping for robotic teams. Journal of Field Robotics **36**(4), 763–781 (2019) 4
38. Schonberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4104–4113 (2016) 2, 3
39. Snavely, N., Seitz, S.M., Szeliski, R.: Photo tourism: exploring photo collections in 3d. In: ACM siggraph 2006 papers, pp. 835–846 (2006) 2, 3
40. Sucar, E., Liu, S., Ortiz, J., Davison, A.J.: imap: Implicit mapping and positioning in real-time. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 6229–6238 (2021) 4
41. Sun, L.C., Bhatt, N.P., Liu, J.C., Fan, Z., Wang, Z., Humphreys, T.E., Topcu, U.: Mm3dgs slam: Multi-modal 3d gaussian splatting for slam using vision, depth, and inertial measurements. In: 2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 10159–10166. IEEE (2024) 4
42. Teed, Z., Deng, J.: Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras. Advances in neural information processing systems **34**, 16558–16569 (2021) 2, 3
43. Tosi, F., Zhang, Y., Gong, Z., Sandström, E., Mattoccia, S., Oswald, M.R., Poggi, M.: How nerfs and 3d gaussian splatting are reshaping slam: a survey. arXiv preprint arXiv:2402.13255 **4**, 1 (2024) 2, 3
44. Wang, F., Zhu, Q., Chang, D., Gao, Q., Han, J., Zhang, T., Hartley, R., Pollefeys, M.: Learning-based multi-view stereo: A survey. arXiv preprint arXiv:2408.15235 (2024) 3
45. Wang, H., Agapito, L.: 3d reconstruction with spatial memory. In: 2025 International Conference on 3D Vision (3DV). pp. 78–89. IEEE (2025) 4
46. Wang, H., Wang, J., Agapito, L.: Co-slam: Joint coordinate and sparse parametric encodings for neural real-time slam. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13293–13302 (2023) 4
47. Wang, J., Chen, M., Karaev, N., Vedaldi, A., Rupprecht, C., Novotny, D.: Vggt: Visual geometry grounded transformer. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 5294–5306 (2025) 3, 4, 5
48. Wang, S., Leroy, V., Cabon, Y., Chidlovskii, B., Revaud, J.: Dust3r: Geometric 3d vision made easy. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 20697–20709 (2024) 2, 4, 8, 10, 12, 13, 14

49. Yan, C., Qu, D., Xu, D., Zhao, B., Wang, Z., Wang, D., Li, X.: Gs-slam: Dense visual slam with 3d gaussian splatting. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 19595–19604 (2024) 4
50. Yang, X., Li, H., Zhai, H., Ming, Y., Liu, Y., Zhang, G.: Vox-fusion: Dense tracking and mapping with voxel-based neural implicit representation. In: 2022 IEEE International Symposium on Mixed and Augmented Reality (ISMAR). pp. 499–507. IEEE (2022) 4
51. Yeshwanth, C., Liu, Y.C., Nießner, M., Dai, A.: Scannet++: A high-fidelity dataset of 3d indoor scenes. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 12–22 (2023) 10
52. Yugay, V., Gevers, T., Oswald, M.R.: Magic-slam: Multi-agent gaussian globally consistent slam. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 6741–6750 (2025) 2, 4, 9, 10, 11, 13, 14
53. Zhang, Y., Tosi, F., Mattoccia, S., Poggi, M.: Go-slam: Global optimization for consistent 3d instant reconstruction. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3727–3737 (2023) 4
54. Zhou, Y., Wu, H.: Multi-agent monocular dense slam with 3d reconstruction priors. arXiv preprint arXiv:2511.19031 (2025) 10, 13, 14
55. Zhu, L., Li, Y., Sandström, E., Huang, S., Schindler, K., Armeni, I.: Loopsplat: Loop closure by registering 3d gaussian splats. In: 2025 International Conference on 3D Vision (3DV). pp. 156–167. IEEE (2025) 4
56. Zhu, Z., Peng, S., Larsson, V., Xu, W., Bao, H., Cui, Z., Oswald, M.R., Pollefeys, M.: Nice-slam: Neural implicit scalable encoding for slam. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 12786–12796 (2022) 4