2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

GRS-SLAM3R: Real-Time Dense SLAM with Gated Recurrent State

Guole Shen[1,*], Tianchen Deng[1,*], Yanbo Wang[1], Yongtao Chen[1], Yilin Shen[1], Jiuming Liu[1], Jingchuan Wang[1]
[1]Shanghai Jiao Tong University

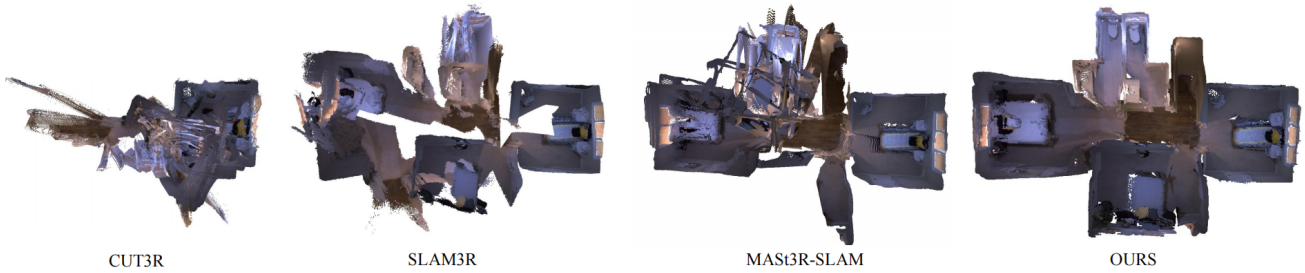


Fig. 1: Scene Reconstruction Performance. We demonstrate the effectiveness of our method in a large-scale, multi-room apartment scene ($\sim 100\,\mathrm{m}^2$). Our approach outperforms existing DUSt3R-based methods in terms of reconstruction quality and completeness.

***Abstract*— DUSt3R-based end-to-end scene reconstruction has recently shown promising results in dense visual SLAM. However, most existing methods only use image pairs to estimate pointmaps, overlooking spatial memory and global consistency. To this end, we introduce GRS-SLAM3R, an end-to-end SLAM framework for dense scene reconstruction and pose estimation from RGB images without any prior knowledge of the scene or camera parameters. Unlike existing DUSt3R-based frameworks, which operate on all image pairs and predict per-pair point maps in local coordinate frames, our method supports sequentialized input and incrementally estimates metric-scale point clouds in the global coordinate. In order to improve consistent spatial correlation, we use a latent state for spatial memory and design a transformer-based gated update module to reset and update the spatial memory that continuously aggregates and tracks relevant 3D information across frames. Furthermore, we partition the scene into submaps, apply local alignment within each submap, and register all submaps into a common world frame using relative constraints, producing a globally consistent map. Experiments on various datasets show that our framework achieves superior reconstruction accuracy while maintaining real-time performance.**

I. INTRODUCTION

Dense visual SLAM has been a long-standing challenge in computer vision, aiming to reconstruct the scene and estimate camera poses directly from image inputs. Over the years, several traditional methods [1], [2], [3] have been developed, relying on handcrafted descriptors for image matching and representing scenes with sparse feature point maps. However, the sparsity of these maps makes it difficult for humans to interpret how the system perceives and interacts with the environment. Moreover, such representations fall short of the requirements for tasks like collision avoidance and motion planning. To address these limitations, research has shifted toward dense scene reconstruction, as demonstrated by systems like DTAM [4] and Kintinuous [5]. Despite their progress, these dense methods often suffer from high memory consumption and slow processing speeds, limiting their applicability in real-time and large-scale scenarios. Researchers then focus on integrating implicit scene representation [6] and 3D Gaussian Splatting [7] with SLAM systems, leading to the emergence of NeRF-based SLAM methods [8], [9], [10], [11], [12] and 3DGS-based SLAM methods [13], [14], [15], [16], [17]. However, all these methods rely on accurate camera intrinsics and depth image inputs, which are often difficult to obtain in real-world scenarios. Meanwhile, 3D scene representation is shifting toward foundation models as the dominant paradigm [18].

DUSt3R [19] shows unprecedented performance and generalization across various real-world scenarios. It operates on image pairs and uses a global alignment method to align the predicted pointmap into the global map. Some works like CUT3R [20] and Spann3R [21] further improve the framework with a continuous update state and an external memory database. With the introduction of DUSt3R [19], researchers have begun to combine the DUSt3R framework into dense SLAM. SLAM3r [22] and MASt3R-SLAM [23] is the pioneer work of DUSt3R-based SLAM. Specifically, SLAM3R proposes a multi-frame registration framework consisting of the Image-to-Points (I2P) network and the Local-to-World (L2W) network, while MASt3R-SLAM leverages the prior from MASt3R and introduces a feature matching based SLAM pipeline. However, these approaches overlook the

Guole Shen, Tianchen Deng, Yanbo Wang, Yongtao Chen, Yilin Shen, Jiuming Liu, Jingchuan Wang are with the Institute of Medical Robotics, School of Automation and Intelligent Sensing, Shanghai Jiao Tong University, Shanghai 200240, Key Laboratory of System Control and Information Processing, Ministry of Education of China, Shanghai 200240. The first two authors contribute equally to this paper. (*corresponding author: jchwang@sjtu.edu.cn)

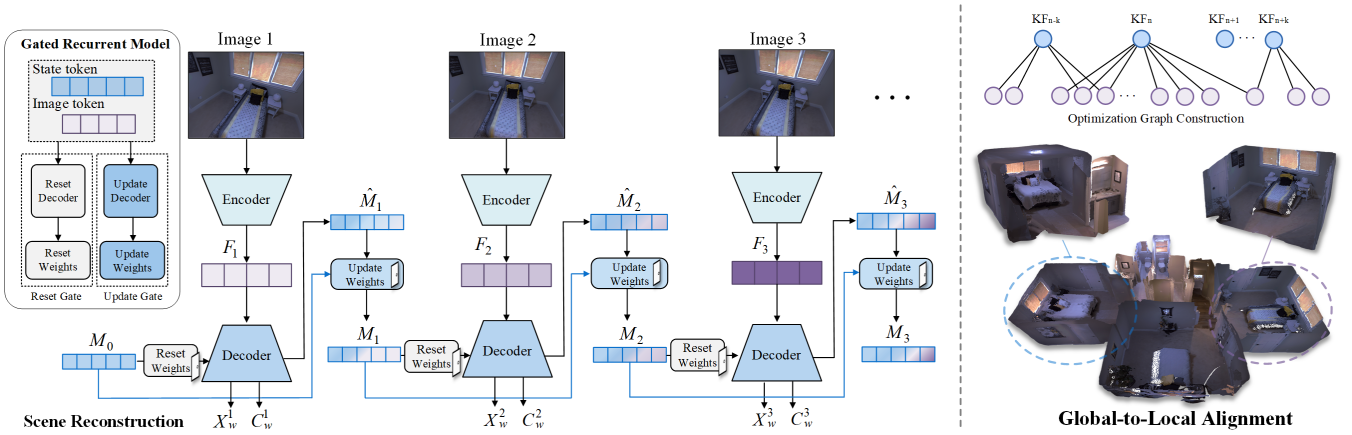


Fig. 2: **System overview.** The input to our system is a stream of RGB images, and the output of the GRS-SLAM3R system consists of camera poses and dense point clouds. On the left side of the figure, we illustrate our scene representation based on a gated recurrent mechanism. Each input frame is encoded into image tokens via an encoder, which interact with a latent state M_t that is updated through reset and update gates. The updated state $\hat{M}_t$ is then decoded to produce the corresponding pose and point cloud in the world coordinate system. On the right side, we present the multi-submap scene representation.

importance of multi-frame spatial correlation and spatial memory that are crucial for large-scale and long-sequence scene reconstruction.

To address this, we propose an incremental SLAM framework that introduces a latent state to enable metric-scale point cloud estimation, with all reconstructions consistently aligned in the world coordinate system. Importantly, directly updating the latent state with each incoming frame may introduce accumulated drift and noise, particularly under significant viewpoint changes. To mitigate this, we design a gated recurrent mechanism and innovatively introduce two transformer-based gating units: an update gate and a reset gate, which respectively select the relevant information from the current frame, and discard the irrelevant information from the memory. Furthermore, we use a keyframe-based submap representation with per-submap state reset to reduce drift; inter-submap registration and local refinement aligns submaps while preserving global consistency and local accuracy. **Overall, our contributions are shown as follows:**

- We propose GRS-SLAM3R, a novel end-to-end incremental dense SLAM framework with gated recurrent model and hierarchical submap alignment, achieving accurate scene reconstruction and pose estimation.
- A novel latent state with gated recurrent model is proposed for consistent spatial correlation. The gated recurrent model includes two transformer-based gates for updating and resetting the memory.
- We propose a multi-submap scene representation with hierarchical alignment: intra-submap local refinement and inter-submap registration that stitches submaps into a coherent map. Experiments on various datasets demonstrate the superiority of the proposed method in both mapping and tracking.

II. Related Work

A. Dense Monocular SLAM

Dense monocular SLAM aims to reconstruct geometry using only RGB images, avoiding the need for depth sensors. Early methods optimize photometric consistency between frames to estimate depth and pose [4], [24], but were limited by drift, noise, and incomplete reconstructions. To enhance robustness and completeness, later approaches incorporate learned priors for depth and visual features [25], [26], combining data-driven inference with geometric optimization. This hybrid strategy improves reconstruction quality but introduces runtime overhead and generalization issues. More recent systems tightly couple learning and optimization [27], [28], [29], achieving better accuracy, though dense monocular SLAM still struggles with real-time performance and global consistency in unconstrained environments.

B. 3D Reconstruction

Dense 3D reconstruction remains a fundamental yet challenging task in computer vision. Traditional methods follow a sequential pipeline: Structure-from-Motion (SfM) [30], [31], [32], [33], [34], [35] recovers sparse geometry and camera poses through keypoint detection, matching, triangulation, and bundle adjustment. Multi-View Stereo (MVS) [36], [37], [38], [39], [40] estimates dense geometry from known poses. While recent works improve individual components with learned features [41], neural bundle adjustment [42], and explore depth prediction [43], [44], [45], [46] as an alternative means to enhance geometry estimation, these pipelines remain sensitive to early-stage errors and typically require camera calibration and offline optimization. Neural rendering methods like NeRF [6] and 3D Gaussian Splatting [7] offer high fidelity but still rely on known poses and per-scene optimization.

C. DUSt3R-Based Online Continuous 3D Reconstruction

DUSt3R [19] introduces a groundbreaking paradigm shift by directly regressing a pointmap, a standard representation

in visual localization, from a pair of images without relying on any prior knowledge of the scene. This approach challenges conventional assumptions and opens up new possibilities for dense reconstruction from minimal input. Several recent works extend DUSt3R's pointmap-based reconstruction to online and continuous settings. Spann3R [21] uses an external spatial memory to perform incremental scene reconstruction in a unified coordinate system. CUT3R [20] incorporates recurrent states for sequential integration. However, these methods often suffer from drift and geometric inconsistency due to the lack of global correction. SLAM3R [22] and MASt3R-SLAM [23] are the most relevant SLAM systems to our work. SLAM3R performs incremental mapping through the Local-to-World (L2W) network, while MASt3R-SLAM builds on the MASt3R [47] two-view prior and integrates efficient point map matching, tracking, fusion, and global optimization. However, both methods overlook the spatial memory and multi-frame correlation, resulting in a lack of consistency during the mapping process.

III. Method

In this paper, we propose GRS-SLAM3R, a novel monocular RGB SLAM system for high-quality online dense 3D reconstruction.

Our system processes a sequence of monocular RGB images $\{I_t \in \mathbb{R}^{H\times W\times 3}\}_{t=1}^{N}$ without relying on external pose or depth supervision. The output of our system is the dense 3D pointcloud $X_t \in \mathbb{R}^{M\times 3}$, where M is the number of 3D points, the corresponding confidence C_t, and the pose P_t. We show the overview of our system in Fig. 2. We design a persistent latent state representation with gated update model for spatial correlation and long-range memory (Sec. III-A). Furthermore, We use a submap-reset representation that updates the latent state locally, preventing state degradation and reducing long-range drift. To integrate multiple submaps, we adopt inter-submap registration and intra-submap refinement.(Sec. III-B).

A. Scene Reconstruction

Existing approaches like DUSt3R [19] operate on pairs of images, which limits their scalability and suitability for real-time, incremental reconstruction. To address this, we propose an incremental scene representation framework with a latent state. Specifically, we design a gated update mechanism that employs two transformer-based gates to effectively update the latent state of the current frame while discarding irrelevant information, thereby preventing the introduction of noise into the memory. In contrast to pairwise predictions in DUSt3R [19], our method can directly output point maps in the world coordinate, which better fits the requirements of online SLAM systems.

1) Gated Recurrent Model: At each time step t, the system receives an input image I_t, which is first encoded into token representations F_t through a vision transformer encoder [48]:

$$F_t = \text{Encoder}(I_t) \tag{1}$$

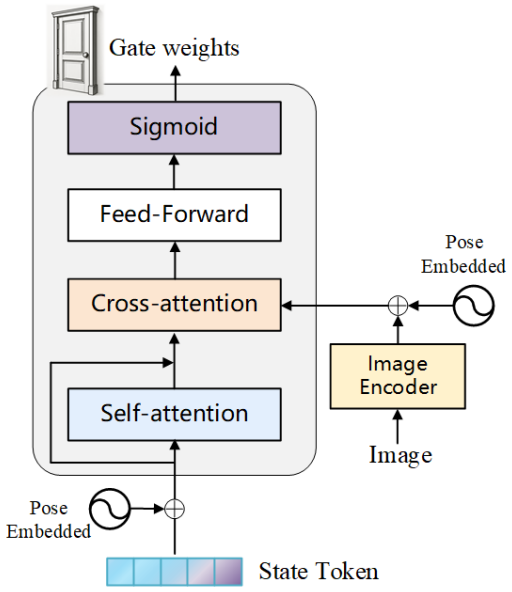


Fig. 3: **Gate structure**. We illustrate the detailed structures of the reset and update gates we designed. Both gates take the historical latent state and the current frame as inputs, and compute the gating weights through a combination of self-attention and cross-attention mechanisms.

Inspired by CUT3R [20], we incorporate a latent memory state M_t that progressively accumulates scene information across sequential observations. We represent the latent state as a set of tokens.

However, directly integrating new image features F_t into the state without regulation can result in drift, the introduction of noise, and contamination of long-term memory, which becomes particularly problematic in long sequences. To address these challenges, we design a gated update mechanism inspired by the traditional GRU structure [49], which selectively integrates new information while preserving existing memory. The image tokens F_t will interact with the memory in two directions. We update the state with information from the current image feature, then we retrieve contextual cues from the state to incorporate knowledge accumulated from past frames. This mechanism ensures stable state updates and accurate information propagation, enabling effective spatial correlation across multiple frames and consistent scene reconstruction over long sequences.

To this end, we design two transformer-based gates in the decoder: a reset gate $G_r(\cdot)$ and an update gate $G_u(\cdot)$. The reset gate determines how the current input information is combined with the previous memory, while the update gate controls how much of the previous memory is preserved and carried over to the current time step. The detailed structure of the gates is shown in Fig. 3. The input of the reset gate $G_r(\cdot)$ is image tokens of the current frame F_t and the memory state of the last time step M_{t-1}. The output of the reset gate U_t determines which parts of the previous memory M_{t-1} should be suppressed before incorporating new observations, thereby regulating the influence of outdated or irrelevant

information. The gates can be formulated as:

$$R_t = G_r(M_{t-1}, F_t), \quad U_t = G_u(M_{t-1}, F_t) \tag{2}$$

The input of update gate $G_u(\cdot)$ is the same as the reset gate. The update gate controls how much information M_{t-1} should be carried forward, determining the extent to which the memory from the previous timestep is preserved. The output of the reset gate G_r is then applied element-wise to the memory:

$$M_t^{\text{reset}} = R_t \odot M_{t-1} \tag{3}$$

Then, we obtain the reset memory M_t^{reset} where irrelevant or outdated information is discarded. To integrate the updated memory with current observations, we design a transformer-based decoder, in which the memory tokens interact with the current frame features F_t and the pose token z_t:

$$[\hat{M}_t, z_t' \oplus F_t'] = \text{Decoder}\left([M_t^{\text{reset}}, z_t \oplus F_t]\right) \tag{4}$$

where $\hat{M}_t$ denotes updated memory state, F_t' are the enriched frame features, and z_t' is a dedicated pose token capturing global scene-level information. The decoder applies cross-attention between memory tokens and image tokens at each block, enabling bidirectional information flow across memory, image, and pose tokens.

The final memory state is then updated via gated update:

$$M_t = U_t \odot \hat{M}_t + (1 - U_t) \odot M_{t-1} \tag{5}$$

This two-stage gating process enables our system to selectively suppress unreliable memory content while adaptively integrating informative new observations, enhancing long sequence consistency and robustness.

After the gated memory update, explicit metric-scale 3D pointclouds can be extracted from z_t' and F_t' for each frame. Specifically, we can generate dense 3D point clouds in both the local camera frame and the global world frame, along with 6-DoF camera poses:

$$\hat{X}_t^{\text{self}}, \hat{C}_t^{\text{self}} = \text{Head}_{\text{self}}(F_t') \tag{6}$$

$$\hat{X}_t^{\text{world}}, \hat{C}_t^{\text{world}} = \text{Head}_{\text{world}}(F_t', z_t') \tag{7}$$

$$\hat{P}_t = \text{Head}_{\text{pose}}(z_t') \tag{8}$$

where $\text{Head}_{\text{self}}(\cdot)$ and $\text{Head}_{\text{world}}(\cdot)$ are implemented as DPT [50], and $\text{Head}_{\text{pose}}(\cdot)$ is implemented as an MLP network, respectively. All pointmaps and poses are in metric scale.

B. Hierarchical Submap Alignment

While the gated memory mechanism facilitates the selection of historical memories, it suffers from accumulated drift in long-sequence. To address this issue, we propose a multi-submap scene representation method, where submaps are segmented based on changes in keyframe viewpoints. We adopt a hierarchical alignment scheme: local alignment within each submap and inter-submap registration for global consistency. This two-level process limits error accumulation over long sequences and preserves fine-grained geometric accuracy, improving cross-submap coherence.

a) Frontend Submap Construction and Keyframe Selection: As each new frame I_t arrives, we evaluate covisibility with the last keyframe I_k and the current submap's anchor I_{a_s} (the first frame of submap S_s). If $\text{Cov}(I_t, I_k) < \tau_{\text{kf}}$, we promote I_t to a keyframe within S_s. If $\text{Cov}(I_t, I_{a_s}) < \tau_{\text{anchor}}$, we start a new submap S_{s+1}, reset the latent state, and set I_t as its anchor. Each submap maintains its own latent memory and local coordinate frame. This submap–reset policy bounds error accumulation within a submap and prevents long-horizon drift from propagating through the recurrent state.

b) Local Submap Alignment: We perform local alignment within each submap. We construct a local connectivity graph $\mathcal{G}_l = (\mathcal{V}_l, \mathcal{E}_l)$ for each submap. The vertex set $\mathcal{V}_l$ includes all frames in the submap, and the edge set $\mathcal{E}_l$ is built by treating each keyframe $k \in \mathcal{K}_l$ as a center and connecting it with its temporally adjacent keyframes and ordinary frames within a fixed window. Specifically, for a keyframe n, we define a local neighbor $\{n-k, \ldots, n+k\}$ and construct pairwise edges with all frames in this group. Each edge is associated with predicted dense pointmaps and confidence maps, and contributes to the local alignment loss. We use local alignment to register all frames within each submap to the world coordinate system, jointly optimizing the pose and point cloud of each frame.

$$\mathcal{L}_{\text{local}}^S = \sum_{e \in \mathcal{E}_l^S} \sum_{v \in e} \sum_{i=1}^{HW} C_i^{v,e} \left\| \xi_i^{v,S} - \sigma_e^S P_e^S X_i^{v,e} \right\| \tag{9}$$

where $\xi^{v,S}$ denotes the optimized local pointmap for keyframe v in submap S, P_e^S is the per-edge rigid transform that maps $X^{v,e}$ to the submap coordinate, and $\sigma_e^S > 0$ is an optional per-edge scale. Minimizing $\mathcal{L}_{\text{local}}^S$ jointly refines the poses and pointmaps of frames within S, producing a self-consistent local reconstruction.

c) Inter-Submap Alignment: We construct a pose graph with submap poses $T_s \in \text{SE}(3)$ as nodes and edges from adjacent-submap alignments and loop closures, Because all submaps share a common metric scale, the inter-submap constraint is a pure rigid transform $\Delta T_{s,s+1} \in \text{SE}(3)$. When S_s is finalized, we query all previous keyframes and retain the top-ranked loop hat exceeds the score threshold, yielding additional relative constraints $\Delta T_{i,j}$. and then solve the resulting pose-graph optimization.

$$\min_{\{T_s\}} \sum_{(u,v) \in \mathcal{E}} \left\| \text{Log}(\Delta T_{u,v}^{-1} T_u^{-1} T_v) \right\|_{\Sigma}^2 + \left\| \text{Log}(T_{s_0}^{-1} \bar{T}_{s_0}) \right\|_{\Sigma_0}^2 \tag{10}$$

where the small-noise prior $(s_0, \bar{T}_{s_0}, \Sigma_0)$ fixes the global gauge. Constraints are added incrementally, adding a loop appends a factor and triggers Levenberg–Marquardt re-optimization.

C. Training loss

Given a sequence of N images, we follow CUT3R [20] and supervise with a confidence-weighted 3D regression loss and a pose loss.

a) Regression loss: We supervise dense pointmap prediction with a scale-aware, confidence-weighted loss. Let x_i be the ground-truth 3D point and $\hat{x}_i$ the prediction at pixel i (confidence $c_i \in [0,1]$). Denoting normalization factors by s (GT) and $\hat{s}$ (prediction), we define

$$\mathscr{L}_{\text{regr}} = \sum_{i=1}^{M} \Big(c_i \big\| \tfrac{\hat{x}_i}{\hat{s}} - \tfrac{x_i}{s} \big\|_2 - \beta \log c_i \Big). \tag{11}$$

When the ground-truth pointmaps are already in metric scale, we set $\hat{s} = s$ to remove scale ambiguity.

b) Pose loss: Let the predicted pose be $\hat{P}_t = (\hat{q}_t, \hat{\tau}_t)$ with quaternion $\hat{q}_t$ and translation $\hat{\tau}_t$ at time t; (q_t, τ_t) are the ground-truths. We minimize rotational and translational discrepancies (the latter normalized consistently with the regression loss):

$$\mathscr{L}_{\text{pose}} = \sum_{t=1}^{N} \Big(\|\hat{q}_t - q_t\|_2 + \big\| \tfrac{\hat{\tau}_t}{\hat{s}} - \tfrac{\tau_t}{s} \big\|_2 \Big). \tag{12}$$

c) Curriculum training: We resize the longer image side to 512 and adopt a three-stage curriculum for stable convergence. We adopt a three-stage curriculum: first, we train on 4-frame sequences, freezing the encoder and part of the decoder and updating only the two gating modules; second, we keep the encoder frozen, unfreeze the decoder, and train the decoder together with the gating modules; finally, we extend sequences to 64 frames and fine-tune the decoder, gating modules, and prediction heads.The latter stages strengthen inter-frame reasoning and improve long-range spatial modeling. We train our network on a various set of 10 datasets, covering synthetic and real-world data, scene-level and object-centric scenes, as well as both indoor and outdoor scenes. Examples of our datasets include CO3Dv2 [51], ARKitScenes [52], ScanNet [53], WildRGBD [54], BlendedMVS [55], Matterport3D [56].

IV. Experiments

A. Experimental Setup

We evaluate our method in terms of surface reconstruction quality, camera pose estimation accuracy, and real-time performance.

a) Test Datasets: We evaluate the effectiveness of our method on small-scale real-world scenes using NRGBD [57] and a subset of 18 sequences from 7-Scenes [58]. To further demonstrate performance in long sequences and large-scale scenarios, we evaluate on the Apartment dataset [59] (multi-room, $\sim 100\,\text{m}^2$) and on the NES dataset [60] [12]. NES spans $> 1000\,\text{m}^2$ with a total trajectory length of 1,482.75 m.

b) Evaluation Metrics: We use absolute trajectory error (ATE-RMSE) to evaluate camera tracking. We evaluate reconstruction quality using accuracy and completeness. Following NICER-SLAM [61], Spann3R [21], and SLAM3R [22], we generate ground-truth point clouds by projecting depth maps into 3D using known camera intrinsics and poses for each test sequence. To account for potential scale discrepancies across methods, we adopt the alignment strategy of SLAM3R [22]: a global similarity transform estimated via the Umeyama algorithm, followed by ICP refinement to minimize residual geometric error.

Method	Chess	Fire	Heads	Office	Pump.	RedKit.	Stairs	Avg.
CUT3R [20]	5.90	5.34	6.37	13.85	14.73	9.44	**6.67**	8.90
MASt3R-SLAM [23]	7.24	5.78	**3.68**	**13.31**	**12.87**	10.07	6.68	8.52
Ours	**5.30**	**5.31**	4.09	13.43	14.41	**8.95**	6.81	**8.27**

TABLE I: Qualitative pose estimation runtime on 7Scenes dataset [58]. We report per-scene ATE RMSE in centimeters.

Method	Acc.[cm]		Comp.[cm]	
	Mean	Median	Mean	Median
DUSt3R-GA [19]	0.144	**0.019**	0.154	**0.018**
MASt3R-GA [47]	**0.085**	0.033	**0.063**	0.028
Spann3R [21]	0.416	0.323	0.417	0.285
CUT3R [20]	0.099	0.031	0.076	0.026
Ours	0.089	0.030	0.072	0.025

TABLE II: Reconstruction results on the NRGBD dataset [57].

c) Implementation Detail: All the training experiments are conducted on eight NVIDIA A100 GPUs with 80 GB of memory each. The inference and SLAM experiments are conduct on a RTX 4090 GPU.

B. Camera Pose Estimation

Our method demonstrates robust and competitive pose estimation performance across both small- and large-scale scenes, particularly when compared with existing DUST3R-based online approaches. On smaller indoor datasets (e.g., 7-Scenes), the gated recurrent update stabilizes per-frame poses and consistently lowers ATE (Table I). In challenging large-scale, multi-room scenes such as Apartment, shown in Table IV, existing methods suffer from drift or trajectory collapse under long-range motion or abrupt camera rotation. Figure 1 illustrates that CUT3R [20] fails to reconstruct the scene, SLAM3R [22] and MASt3R-SLAM [23] suffer from significant pose drift in certain rooms, whereas our method maintains consistent and stable tracking throughout the scene. Our multi-submap scene representation method bounds the drift and reduces the risk of accumulated drift over long sequences. As shown in Fig. 5, on complex indoor sequences, CUT3R [20] drifts after sharp turns, SLAM3R [22] breaks in long corridors, and MASt3R-SLAM [23] exhibits significant scale variation, with the reconstructed scene size abruptly decreasing after a certain sequence length. In comparison, our method is able to reconstruct corridor structures with stable scale and spatial consistency.

C. Surface Reconstruction

Our method achieves high-quality scene reconstruction across different datasets. We follow the [20] setting to evaluate performance on sparsely sampled images from the NRGBD dataset, which contains minimal or no view overlap, as shown in Table II. The results indicate that the recurrent mechanism preserves key geometry across frames and enables spatial reasoning by selectively integrating structural cues, even with minimal overlap. According to the results

Method	Chess Acc. / Comp.	Fire Acc. / Comp.	Heads Acc. / Comp.	Office Acc. / Comp.	Pumpkin Acc. / Comp.	RedKitchen Acc. / Comp.	Stairs Acc. / Comp.	Average Acc. / Comp.	FPS
DUSt3R [19]	2.26 / 2.13	1.04 / 1.50	1.66 / 0.98	4.62 / 4.74	**1.73** / 2.43	1.95 / 2.36	3.37 / 10.75	2.19 / 3.24	< 1
MASt3R [47]	2.08 / 2.12	1.54 / 1.43	**1.06** / 1.04	3.23 / 3.19	5.68 / 3.07	3.50 / 3.37	2.36 / 13.16	3.04 / 3.90	< 1
Spann3R [21]	2.23 / 1.68	0.88 / 0.92	2.67 / 0.98	5.86 / 3.54	2.25 / **1.85**	2.68 / **1.80**	5.65 / 5.15	3.42 / 2.41	> 50
CUT3R [20]	2.46 / 1.99	1.52 / 1.43	2.10 / 1.13	3.81 / 3.05	2.98 / 2.48	2.49 / 2.24	3.35 / 10.53	2.67 / 3.27	~ 20
SLAM3R [22]	1.63 / **1.31**	**0.84** / **0.83**	2.95 / 1.22	**2.32** / **2.26**	1.81 / 2.05	**1.84** / 1.94	4.19 / 6.91	2.13 / 2.34	~ 25
MASt3R-SLAM [23]	2.41 / 1.70	1.57 / 1.33	1.71 / 1.16	3.47 / 2.98	2.86 / 2.37	2.83 / 2.16	3.32 / 9.53	2.60 / 3.03	~ 15
Ours	**1.49** / 1.32	1.26 / 1.32	1.22 / **0.83**	4.17 / 3.41	2.27 / 2.25	2.19 / 2.19	**2.22** / **4.55**	**2.12** / **2.27**	~ 15

TABLE III: **Reconstruction results and runtime (FPS) on 7Scenes [58]**. We report Accuracy and Completion in centimeters.

Fig. 4: **Qualitative scene reconstruction results.** We demonstrate the mapping performance of our method on chess seq-05 in 7scenes [58]. The region outlined on the image is marked in red to signify lower predictive accuracy, in green to signify higher accuracy, and in yellow to represent the ground truth results.

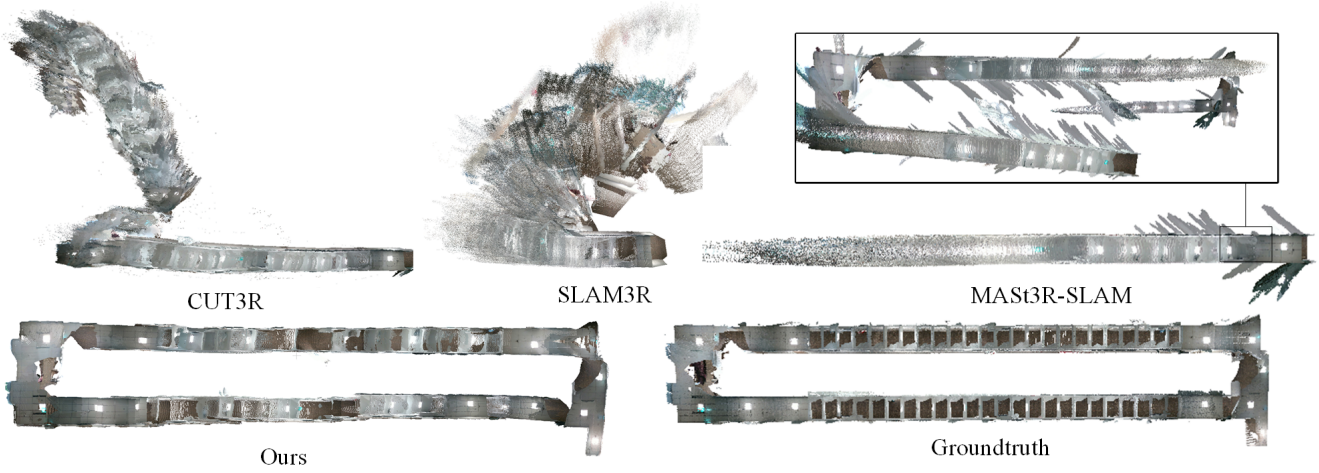


Fig. 5: **Qualitative scene reconstruction results.** We demonstrate the mapping performance of our method on the NES dataset. For MASt3R-SLAM [23], due to the scale reduction in the latter part of the sequence, we provide an enlarged view of this region alongside the full-scene overview. The boxed region highlights the scale-reduced area.

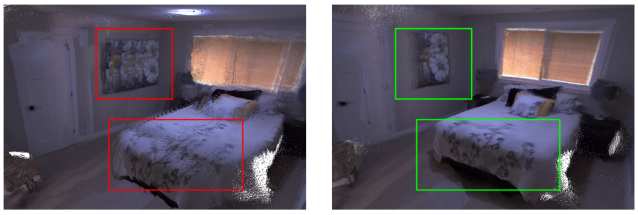
Method	**Acc.**[cm] ↓	**Comp.**[cm] ↓	**ATE**[m] ↓
CUT3R [20]	48.49	39.85	2.39
SLAM3R [22]	21.67	27.66	–
MASt3R-SLAM [23]	8.72	5.80	0.72
Ours	**6.79**	**5.62**	**0.16**

TABLE IV: Quantitative results on the Apartment dataset [59]. ATE is not reported for SLAM3R [22] as it does not produce explicit camera pose estimates during inference.

in Table III, our approach maintains stable and high-quality reconstructions in small indoor scenes such as 7Scenes. Spann3R [21] and CUT3R[20] both incorporate spatial memory mechanisms, similar to our approach. However, they do not effectively address the challenges of memory update and forgetting, which are crucial for spatial memory consistency and robustness. Our recurrent model helps maintain geometric consistency across frames, while local refinement improves surface alignment and enhances fine-grained details. In large-scale scenes, as shown in Table IV, Fig. 1 and Fig. 5, our method achieves accurate and consistent reconstruction, which indicates the effectiveness of our framework. Compared with MASt3R-SLAM [23] and SLAM3R [22], our framework builds consistent spatial multi-view correlations through the gated recurrent model, and further enhances reconstruction consistency via hierarchical alignment.

D. Time Analysis

In Table III, we present the runtime analysis of our framework and other methods. Our method runs at approximately



Ours (w/o gate , w/o Local Align) Ours (with gate , w/o Local Align)

Fig. 6: Ablation on Gated Recurrent model. We conduct an ablation study on the Apartment dataset [59] to evaluate the effectiveness of the gated recurrent state. The gate update module significantly enhances the accuracy and consistency of scene reconstruction.

Modules			Reconstruction (cm)		ATE (m)
Gate	Local Align	Submap	Acc.	Comp.	
✗	✗	✗	48.49	39.85	2.39
✗	✓	✓	8.87	6.30	0.33
✓	✓	✗	28.95	24.93	1.38
✓	✗	✓	7.32	6.18	0.23
✓	✓	✓	**6.79**	**5.62**	**0.16**

TABLE V: Ablation of *Gate*, *Local Align*, and *Submap*. ✓/✗indicate module enabled/disabled. Using all three modules yields the best performance.

15 FPS on a single RTX 4090 GPU, achieving a favorable balance between speed and accuracy compared to existing online SLAM baselines.

E. Ablation Study

In this section, we conduct various experiments to verify the effectiveness of our method. Table V illustrates a quantitative evaluation with different settings. We conduct an ablation study by respectively removing the gated update model, the submap scene representation, and the local alignment to prove the effectiveness of our modules. In comparison, both the submap representation and the state update model significantly contribute to the accuracy of scene reconstruction. Moreover, the hierarchical alignment plays a crucial role in enhancing tracking performance and maintaining overall consistency. As shown in Figure 6, we present an ablation study of the gated recurrent model without local alignment. Without the gated model, it fails to effectively regulate its latent state, resulting in the accumulation of inconsistent geometry and a loss of structural coherence. In contrast, our gated update method enables the selective integration of new observations, preserving spatial consistency across frames.

V. Conclusion

In this paper, we present a real-time dense end-to-end SLAM framework that achieves accurate scene reconstruction and camera tracking. We design a persistent latent state that serves as a spatial memory, continuously interacting with the current frame to update both its point cloud and pose, while simultaneously refining the memory itself. To prevent noise accumulation in the spatial memory and to improve the decoding of the current frame's geometry, we introduce transformer-based update and reset gates that selectively control information flow. Furthermore, we propose a multi-submap representation strategy combined with a hierarchical alignment mechanism, which aligns submaps in a globally consistent manner while locally optimizing point clouds and poses within each submap. Experimental results on extensive datasets verify the effectiveness of our method.

References

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

[2] T. Deng, H. Xie, J. Wang, and W. Chen, "Long-term visual simultaneous localization and mapping: Using a bayesian persistence filter-based global map prediction," *IEEE Robotics & Automation Magazine*, vol. 30, no. 1, pp. 36–49, 2023.

[3] H. Xie, T. Deng, J. Wang, and W. Chen, "Robust incremental long-term visual topological localization in changing environments," *IEEE Transactions on Instrumentation and Measurement*, vol. 72, pp. 1–14, 2022.

[4] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison, "Dtam: Dense tracking and mapping in real-time," in *2011 international conference on computer vision*. IEEE, 2011, pp. 2320–2327.

[5] T. Whelan, H. Johannsson, M. Kaess, J. J. Leonard, and J. McDonald, "Kintinuous: Spatially extended kinectfusion," in *RSS Workshop on RGB-D: Advanced Reasoning with Depth Cameras*, 2012.

[6] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," in *ECCV*, 2020.

[7] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, "3d gaussian splatting for real-time radiance field rendering." *ACM Trans. Graph.*, vol. 42, no. 4, pp. 139–1, 2023.

[8] Z. Zhu, S. Peng, V. Larsson, W. Xu, H. Bao, Z. Cui, M. R. Oswald, and M. Pollefeys, "Nice-slam: Neural implicit scalable encoding for slam," in *CVPR*, June 2022, pp. 12 786–12 796.

[9] T. Deng, G. Shen, T. Qin, J. Wang, W. Zhao, J. Wang, D. Wang, and W. Chen, "Plgslam: Progressive neural scene represenation with local to global bundle adjustment," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 19 657–19 666.

[10] T. Deng, Y. Wang, H. Xie, H. Wang, R. Guo, J. Wang, D. Wang, and W. Chen, "Neslam: Neural implicit mapping and self-supervised feature tracking with depth completion and denoising," *IEEE Transactions on Automation Science and Engineering*, pp. 1–1, 2025.

[11] S. Zhu, G. Wang, H. Blum, J. Liu, L. Song, M. Pollefeys, and H. Wang, "Sni-slam: Semantic neural implicit slam," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 21 167–21 177.

[12] T. Deng, G. Shen, X. Chen, S. Yuan, H. Shen, G. Peng, Z. Wu, J. Wang, L. Xie, D. Wang, H. Wang, and W. Chen, "Mcn-slam: Multi-agent collaborative neural slam with hybrid implicit neural scene representation," *arXiv preprint arXiv:2506.18678*, 2025.

[13] C. Yan, D. Qu, D. Wang, D. Xu, Z. Wang, B. Zhao, and X. Li, "Gs-slam: Dense visual slam with 3d gaussian splatting," *arXiv preprint arXiv:2311.11700*, 2023.

[14] N. Keetha, J. Karhade, K. M. Jatavallabhula, G. Yang, S. Scherer, D. Ramanan, and J. Luiten, "Splatam: Splat track & map 3d gaussians for dense rgb-d slam," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 21 357–21 366.

[15] T. Deng, Y. Chen, L. Zhang, J. Yang, S. Yuan, D. Wang, and W. Chen, "Compact 3d gaussian splatting for dense visual slam," *arXiv preprint arXiv:2403.11247*, 2024.

[16] M. Li, S. Liu, H. Zhou, G. Zhu, N. Cheng, T. Deng, and H. Wang, "Sgs-slam: Semantic gaussian splatting for neural dense slam," in *European Conference on Computer Vision*. Springer, 2024, pp. 163–179.

[17] S. Liu, H. Zhou, L. Li, Y. Liu, T. Deng, Y. Zhou, and M. Li, "Structure gaussian slam with manhattan world hypothesis," *arXiv preprint arXiv:2405.20031*, 2024.

[18] T. Deng, Y. Pan, S. Yuan, D. Li, C. Wang, M. Li, L. Chen, L. Xie, D. Wang, J. Wang, J. Civera, H. Wang, and W. Chen, "What is the best 3d scene representation for robotics? from geometric to foundation models," *arXiv preprint arXiv:2512.03422*, 2025.

[19] S. Wang, V. Leroy, Y. Cabon, B. Chidlovskii, and J. Revaud, "Dust3r: Geometric 3d vision made easy," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 20 697–20 709.

[20] Q. Wang, Y. Zhang, A. Holynski, A. A. Efros, and A. Kanazawa, "Continuous 3d perception model with persistent state," *arXiv preprint arXiv:2501.12387*, 2025.

[21] H. Wang and L. Agapito, "3d reconstruction with spatial memory," *arXiv preprint arXiv:2408.16061*, 2024.

[22] Y. Liu, S. Dong, S. Wang, Y. Yin, Y. Yang, Q. Fan, and B. Chen, "Slam3r: Real-time dense scene reconstruction from monocular rgb videos," *arXiv preprint arXiv:2412.09401*, 2024.

[23] R. Murai, E. Dexheimer, and A. J. Davison, "Mast3r-slam: Real-time dense slam with 3d reconstruction priors," *arXiv preprint arXiv:2412.12392*, 2024.

[24] J. Engel, T. Schöps, and D. Cremers, "Lsd-slam: Large-scale direct monocular slam," in *European conference on computer vision*. Springer, 2014, pp. 834–849.

[25] K. Tateno, F. Tombari, I. Laina, and N. Navab, "Cnn-slam: Real-time dense monocular slam with learned depth prediction," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 6243–6252.

[26] M. Bloesch, J. Czarnowski, R. Clark, S. Leutenegger, and A. J. Davison, "Codeslam — learning a compact, optimisable representation for dense visual slam," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[27] Z. Teed and J. Deng, "Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras," *Advances in neural information processing systems*, vol. 34, pp. 16 558–16 569, 2021.

[28] L. Koestler, N. Yang, N. Zeller, and D. Cremers, "Tandem: Tracking and dense mapping in real-time using deep multi-view stereo," in *Conference on Robot Learning*. PMLR, 2022, pp. 34–45.

[29] J. Liu, G. Wang, C. Jiang, Z. Liu, and H. Wang, "Translo: A window-based masked point transformer framework for large-scale lidar odometry," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 2, 2023, pp. 1683–1691.

[30] P. Lindenberger, P.-E. Sarlin, V. Larsson, and M. Pollefeys, "Pixel-perfect structure-from-motion with featuremetric refinement," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 5987–5997.

[31] A. Sameer, "Building rome in a day," *Proc. ICCV, 2009*, 2009.

[32] J. L. Schonberger and J.-M. Frahm, "Structure-from-motion revisited," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 4104–4113.

[33] C. Sweeney, T. Sattler, T. Hollerer, M. Turk, and M. Pollefeys, "Optimizing the viewing graph for structure-from-motion," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 801–809.

[34] L. Pan, D. Baráth, M. Pollefeys, and J. L. Schönberger, "Global structure-from-motion revisited," in *European Conference on Computer Vision*. Springer, 2024, pp. 58–77.

[35] N. Snavely, S. M. Seitz, and R. Szeliski, "Photo tourism: exploring photo collections in 3d," in *ACM siggraph 2006 papers*, 2006, pp. 835–846.

[36] Y. Yao, Z. Luo, S. Li, T. Fang, and L. Quan, "Mvsnet: Depth inference for unstructured multi-view stereo," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 767–783.

[37] Y. Furukawa, C. Hernández, *et al.*, "Multi-view stereo: A tutorial," *Foundations and trends® in Computer Graphics and Vision*, vol. 9, no. 1-2, pp. 1–148, 2015.

[38] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski, "A comparison and evaluation of multi-view stereo reconstruction algorithms," in *2006 IEEE computer society conference on computer vision and pattern recognition (CVPR'06)*, vol. 1. IEEE, 2006, pp. 519–528.

[39] Y. Furukawa and J. Ponce, "Accurate, dense, and robust multiview stereopsis," *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 8, pp. 1362–1376, 2009.

[40] J. L. Schönberger, E. Zheng, J.-M. Frahm, and M. Pollefeys, "Pixelwise view selection for unstructured multi-view stereo," in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III 14*. Springer, 2016, pp. 501–518.

[41] D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superpoint: Self-supervised interest point detection and description," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2018.

[42] C. Tang and P. Tan, "Ba-net: Dense bundle adjustment network," *ICLR*, 2018.

[43] L. Piccinelli, Y.-H. Yang, C. Sakaridis, M. Segu, S. Li, L. Van Gool, and F. Yu, "Unidepth: Universal monocular metric depth estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 10 106–10 116.

[44] S. F. Bhat, R. Birkl, D. Wofk, P. Wonka, and M. Müller, "Zoedepth: Zero-shot transfer by combining relative and metric depth," *arXiv preprint arXiv:2302.12288*, 2023.

[45] Z. Li and N. Snavely, "Megadepth: Learning single-view depth prediction from internet photos," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 2041–2050.

[46] C. Godard, O. Mac Aodha, M. Firman, and G. J. Brostow, "Digging into self-supervised monocular depth estimation," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 3828–3838.

[47] V. Leroy, Y. Cabon, and J. Revaud, "Grounding image matching in 3d with mast3r," in *European Conference on Computer Vision*. Springer, 2024, pp. 71–91.

[48] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[49] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.

[50] R. Ranftl, A. Bochkovskiy, and V. Koltun, "Vision transformers for dense prediction," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 12 179–12 188.

[51] J. Reizenstein, R. Shapovalov, P. Henzler, L. Sbordone, P. Labatut, and D. Novotny, "Common objects in 3d: Large-scale learning and evaluation of real-life 3d category reconstruction," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 10 901–10 911.

[52] A. Dehghan, G. Baruch, Z. Chen, Y. Feigin, P. Fu, T. Gebauer, D. Kurz, T. Dimry, B. Joffe, A. Schwartz, *et al.*, "Arkitscenes: A diverse real-world dataset for 3d indoor scene understanding using mobile rgb-d data." *NeurIPS Datasets and Benchmarks*, vol. 2, no. 6, p. 16, 2021.

[53] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Niessner, "Scannet: Richly-annotated 3d reconstructions of indoor scenes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[54] H. Xia, Y. Fu, S. Liu, and X. Wang, "Rgbd objects in the wild: scaling real-world 3d object learning from rgb-d videos," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 22 378–22 389.

[55] Y. Yao, Z. Luo, S. Li, J. Zhang, Y. Ren, L. Zhou, T. Fang, and L. Quan, "Blendedmvs: A large-scale dataset for generalized multi-view stereo networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 1790–1799.

[56] A. Chang, A. Dai, T. Funkhouser, M. Halber, M. Niessner, M. Savva, S. Song, A. Zeng, and Y. Zhang, "Matterport3d: Learning from rgb-d data in indoor environments," *arXiv preprint arXiv:1709.06158*, 2017.

[57] D. Azinović, R. Martin-Brualla, D. B. Goldman, M. Nießner, and J. Thies, "Neural rgb-d surface reconstruction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 6290–6301.

[58] J. Shotton, B. Glocker, C. Zach, S. Izadi, A. Criminisi, and A. Fitzgibbon, "Scene coordinate regression forests for camera relocalization in rgb-d images," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2013, pp. 2930–2937.

[59] J. Straub, T. Whelan, L. Ma, Y. Chen, E. Wijmans, S. Green, J. J. Engel, R. Mur-Artal, C. Ren, S. Verma, *et al.*, "The replica dataset: A digital replica of indoor spaces," *arXiv preprint arXiv:1906.05797*, 2019.

[60] T. Deng, G. Shen, C. Xun, S. Yuan, T. Jin, H. Shen, Y. Wang, J. Wang, H. Wang, D. Wang, *et al.*, "Mne-slam: Multi-agent neural slam for mobile robots," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 1485–1494.

[61] Z. Zhu, S. Peng, V. Larsson, Z. Cui, M. R. Oswald, A. Geiger, and M. Pollefeys, "Nicer-slam: Neural implicit scene encoding for rgb slam," in *2024 International Conference on 3D Vision (3DV)*. IEEE, 2024, pp. 42–52.