UniSim-SLAM: Feed-Forward SLAM with Unified Sim(3) Optimization

Inha Lee, Dongjae Jeong, Junhee Lee, and Kyungdon Joo†

Ulsan National Institute of Science and Technology (UNIST), Republic of Korea
{epsilon8854,jeong_dongjae,junhee98,kyungdon}@unist.ac.kr

Abstract. Recent geometric foundation models enable feed-forward inference for SLAM, but their predictions are strongly dependent on the input view set, which leads to geometric inconsistencies and trajectory drift when results are chained over long sequences. Online deployment further exposes a trade-off between the low latency of two-view tracking and the constraint richness of multi-view inference. We introduce `UniSim-SLAM`, an integrated system that runs lightweight two-view keyframe tracking in the frontend and performs periodic multi-view submap refinement in the backend. To combine predictions defined in heterogeneous local coordinates with inconsistent scales, we formulate a unified multi-level factor graph on $Sim(3)$ that jointly optimizes global keyframe poses and submap poses. The graph integrates temporal view-to-view odometry edges, view-to-submap bridge edges with depth-statistics scale anchoring, and submap-to-submap tie and scale constraints to enforce consistent similarity relations across submaps. Experiments on TUM RGB-D and 7-Scenes show that `UniSim-SLAM` achieves state-of-the-art accuracy in the uncalibrated setting, reducing trajectory error by 38.5% on TUM RGB-D and 45.9% on 7-Scenes compared to prior best results. Project page: `https://vision3d-lab.github.io/unisim-slam/`.

Keywords: Visual SLAM · Feed-Forward SLAM · $Sim(3)$ Factor Graph

1 Introduction

Recent progress in geometric foundation models is reshaping the design of visual SLAM systems. For example, models such as DUSt3R [40] and VGGT [38] have enabled feed-forward estimation of dense depth maps and relative camera poses, even from uncalibrated image sets. As a result, recent SLAM systems [22, 26, 46] have begun to apply feed-forward models to specific view subsets, such as image pairs or multi-view image clips, treating each inference output as a local reconstruction defined in its own local coordinate, which is subsequently aligned into a global coordinate system.

One practical challenge in feed-forward SLAM is that geometric estimates are conditioned on the input view configuration. Even for the same image, the

† Corresponding author.

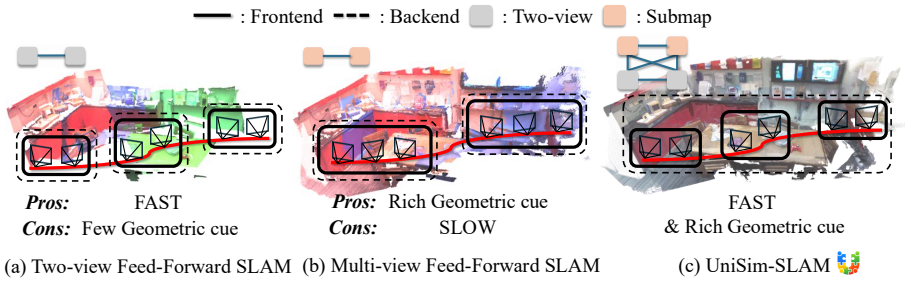

Fig. 1: Trade-off in feed-forward SLAM inference regimes. (a) Two-view inference enables low-latency tracking but provides limited geometric constraints. (b) Multi-view submap inference yields richer geometry but incurs higher latency. (c) `UniSim-SLAM` combines a two-view frontend with a multi-view submap backend and jointly optimizes both in a unified $Sim(3)$ factor graph.

estimated scale and pose can vary depending on the co-visible views, leading to geometric inconsistencies when predictions are chained over long sequences. Accordingly, the global trajectory must be constructed by aligning multiple configuration-dependent local reconstructions, each defined in its own coordinate frame. For instance, two-view approaches align pairwise reconstructions using mutual constraints (*e.g.*, VISTA-SLAM [46]), whereas multi-view methods register submaps by estimating transformations across overlapping regions (*e.g.*, VGGT-SLAM [22]).

Beyond this configuration-dependent inconsistency, relying on a single inference regime (*i.e.*, two-view or multi-view inference) further introduces a fundamental trade-off (see Fig. 1). Multi-view inference uses rich geometric constraints, but it requires accumulating a fixed number of frames (*i.e.*, a submap) before inference, making per-frame updates impractical. Moreover, refinement based solely on relationships between these submaps requires overlap between neighboring submaps to propagate corrections across the trajectory. In particular, when the overlap does not exist, geometric corrections cannot be effectively propagated, limiting global consistency. In contrast, two-view inference is computationally lightweight and operates immediately upon receiving a new frame, resulting in low latency while naturally maintaining temporal connectivity. However, due to its limited geometric constraints by two-view inference, it is prone to drift accumulation over time, making it insufficient for maintaining long-term global consistency. Therefore, relying on either inference regime alone is insufficient to achieve both low-latency tracking and long-term geometric consistency.

To address this trade-off, we revisit the architectural principles of classic SLAM systems. Classic SLAM systems [8,11,24,28] successfully mitigate a similar efficiency-consistency trade-off by combining lightweight two-view odometry in the frontend with intermittent, globally consistent refinement in the backend. This paradigm ensures robust graph connectivity through the temporal consistency of two-view inference while simultaneously using the rich geomet-

ric constraints of multi-view submaps. However, existing feed-forward methods typically perform either pairwise two-view alignment [46] or submap-to-submap registration in isolation [22]. Yet, naively combining two-view and multi-view predictions is non-trivial because their predictions reside in heterogeneous local coordinate systems with inconsistent scales and reference frames. Consequently, a new factor graph formulation is required to jointly optimize these heterogeneous constraints within a single unified framework.

In this work, we propose `UniSim-SLAM`, a new feed-forward SLAM system that unifies two-view and multi-view inferences within a single $Sim(3)$ factor graph (see Fig. 1). Our key insight is that geometric predictions obtained from different feed-forward inference regimes can be interpreted as complementary constraints that reside in heterogeneous local coordinate systems, and therefore must be jointly optimized within a unified $Sim(3)$ factor graph. Specifically, two-view inference provides lightweight and densely connected temporal constraints that support low-latency tracking, while multi-view inference produces geometrically consistent submaps that anchor the global structure of the scene. `UniSim-SLAM` integrates these heterogeneous constraints through a unified multi-level factor graph defined on the $Sim(3)$ manifold, enabling consistent optimization across both frame-level and submap-level predictions. By jointly optimizing two-view and multi-view constraints within this unified framework, `UniSim-SLAM` resolves the efficiency-consistency trade-off in feed-forward SLAM, achieving both low-latency tracking and long-term geometric stability.

We evaluate `UniSim-SLAM` on standard SLAM benchmarks [30, 32], demonstrating state-of-the-art performance. The main contributions are summarized as follows:

- We propose `UniSim-SLAM`, a feed-forward SLAM system that jointly optimizes heterogeneous predictions from two-view and multi-view inference regimes within a unified $Sim(3)$ optimization framework.
- We introduce a unified multi-level factor graph that connects frames and submaps through three complementary edges (*i.e.*, view-to-view, view-to-submap, and submap-to-submap), enabling scale-consistent optimization and drift correction even when submaps do not directly overlap.
- We demonstrate that integrating temporally dense two-view constraints with geometrically rich multi-view submaps enables robust correction propagation across long trajectories, achieving state-of-the-art performance on standard SLAM benchmarks.

2 Related Work

2.1 Visual SLAM

Visual SLAM can be broadly categorized into two paradigms, feature-based methods and direct methods. Feature-based methods such as ORB-SLAM [3, 24, 25] and Kimera [28] follow the SfM pipeline [1, 20, 29], leveraging keypoint matching for triangulation and PnP [15, 16]. In contrast, direct methods [8, 9]

optimize camera poses by minimizing photometric residuals, typically alongside per-frame depth estimation. Despite different frontends, both typically rely on an optimization backend, most commonly bundle adjustment [36]. They are sensitive to camera calibration and challenging visual conditions, often producing only sparse or semi-dense maps.

To overcome these limitations, learning-based SLAM integrates deep neural networks into the frontend or the scene representation to improve robustness and densify mapping. DeepFactors [4] and DeepV2D [34] focus on learned depth and pose with multi-view consistency, whereas DROID-SLAM [35] and DPV-SLAM [19] couple learned correspondences with differentiable bundle adjustment for iterative refinement. Neural implicit mapping jointly optimizes camera poses with an implicit scene representation online. For example, iMAP [33] and NICER-SLAM [49] use MLP-based implicit maps for online reconstruction, while GlORIE-SLAM [47] adopts a deformable neural point-cloud representation with loop closure and online global BA for improved global consistency. More recently, 3D Gaussian Splatting (3DGS) [14] has enabled efficient differentiable dense SLAM. 3DGS-based methods, including Gaussian Splatting SLAM [23] and GS-SLAM [43], jointly optimize poses and Gaussian primitives.

Nevertheless, most learning-based and neural mapping approaches still rely on accurate intrinsics and remain computationally heavy for real-time use.

2.2 Feed-Forward Visual SLAM

Recent progress in large-scale self-supervised representation learning and geometry estimation (*e.g.*, DINOv2 [27], Depth Anything [18, 44, 45], FoundationStereo [42]) has enabled feed-forward 3D geometric foundation models that predict geometry and camera parameters in a feed-forward manner, reducing reliance on handcrafted correspondences and heavy per-sequence optimization. Based on these advances, pairwise feed-forward models [17, 40] regress 3D structure and dense correspondences from uncalibrated image pairs, and can be scaled to unconstrained collections via retrieval-based view graphs and global alignment [7]. Complementary directions include direct relative pose regression [6] and feed-forward prediction of 3D Gaussian primitives without intrinsics [31].

Beyond pairwise settings, these models also support multi-view and long-horizon inference. Memory or state-based models [37, 39] enable incremental reconstruction in a global frame, while long-term tracking provides a robust correspondence backbone [12]. Recent multi-view foundation models further infer dense geometry jointly with camera parameters from one to many views in a single pass (*e.g.*, VGGT [38], π^3 [41], MapAnything [13]), with VGGT-Long [5] improving scalability to long RGB streams via chunk-wise reconstruction and lightweight loop-closure optimization.

Leveraging these models, feed-forward visual SLAM methods have emerged and can be broadly grouped into two-view and multi-view pipelines. Two-view feed-forward SLAM conducts pairwise predictions and lightweight global optimization. For example, MASt3R-SLAM [26] uses MASt3R [17] as a two-view 3D reconstruction and matching prior and builds a real-time dense monocular

SLAM system with pointmap-based matching, local fusion, loop closure, and second-order global optimization. Similarly, ViSTA-SLAM [46] proposes a symmetric two-view association frontend that regresses local point maps and relative pose from two RGB images, mitigating drift via $Sim(3)$ pose-graph optimization with loop closure in the backend. Multi-view feed-forward SLAM focuses on incrementally aligning and globally optimizing submaps produced by multi-view reconstruction backbones. VGGT-SLAM [22] incrementally builds VGGT submaps and globally aligns them by optimizing 15-DoF projective transformations on the $SL(4)$ manifold to handle ambiguity under uncalibrated cameras. Most feed-forward SLAM systems exploit either two-view temporal constraints or multi-view submap constraints in isolation. In contrast, `UniSim-SLAM` jointly utilizes both within a unified multi-level factor graph.

3 Method

3.1 Overview

`UniSim-SLAM` revisits the classical SLAM paradigm to address the trade-off between inference latency and geometric constraint richness in feed-forward SLAM. Our system combines a lightweight two-view frontend for low-latency tracking with a constraint-rich multi-view backend for drift correction. However, two-view and multi-view inferences are defined in different local coordinate systems and exhibit inconsistent scale and reference frames. To coherently integrate these heterogeneous predictions, we formulate a unified pose graph on the $Sim(3)$ manifold, which naturally accommodates rotation, translation, and scale ambiguity (see Fig. 2).

Given an input image stream, the frontend samples keyframes and incrementally estimates global poses using pairwise two-view predictions. In parallel, the backend periodically constructs multi-view submaps over contiguous keyframe windows, producing submap-local pose estimates. The optimization variables of the unified factor graph consist of global keyframe poses $\{T_i\}$ and submap poses $\{S_m\}$, both of which are jointly refined within a unified $Sim(3)$ optimization framework. Details of the frontend and backend are described in Secs. 3.2 and 3.3, followed by the unified factor graph formulation in Sec. 3.4.

3.2 Two-View Tracking on Keyframes

The frontend plays a central role in ensuring low latency and temporal consistency. For each temporally consecutive keyframe pair (I_i, I_j) with $j = i + 1$, we feed the image pair into a feed-forward model (*e.g.*, VGGT [38], STA [46]) to obtain two-view estimations:

$$\{\hat{D}_i^{2\text{v}}, \hat{D}_j^{2\text{v}}, \hat{T}_{ij}^{2\text{v}}\} = f_{2\text{v}}(I_i, I_j), \tag{1}$$

where $\hat{D}_i^{2\text{v}}$ and $\hat{D}_j^{2\text{v}}$ denote the predicted depth maps, and $\hat{T}_{ij}^{2\text{v}} \in Sim(3)$ represents the relative transformation from keyframe I_i to I_j with its scale component

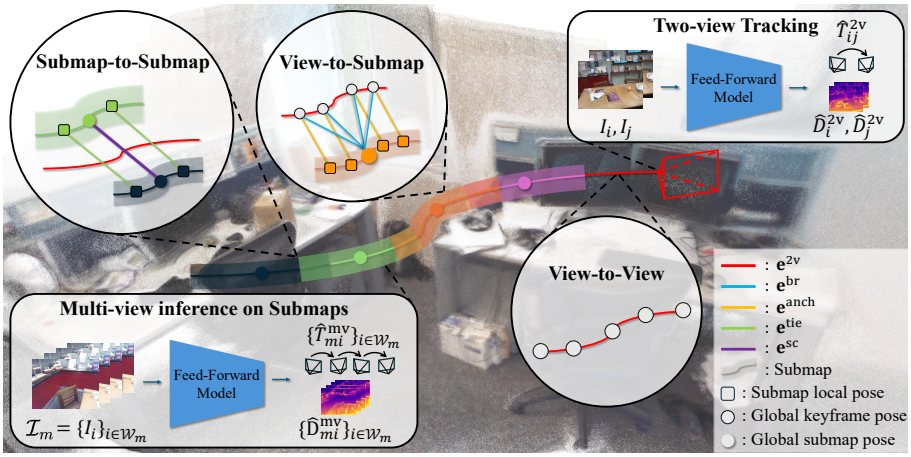

Fig. 2: Overall framework of `UniSim-SLAM`. A feed-forward model produces two-view pose and depth for tracking and multi-view submap predictions for local geometry. These predictions are integrated into a unified $Sim(3)$ factor graph with multiple edge types, jointly optimizing global keyframes and submap poses.

initialized to 1. The predicted depth maps are later used to estimate scale anchors for submap integration.

To construct a global trajectory from these relative measurements, we first define a global reference frame. The first keyframe I_0 defines the origin of the global coordinate system. Then, global poses are initialized online by sequentially composing the two-view relative transformations:

$$T_j = T_i \hat{T}_{ij}^{\mathrm{2v}}. \tag{2}$$

This sequential initialization implicitly defines the temporal edges of the pose graph and maintains its connectivity even when submaps do not overlap. The resulting trajectory serves as the initial estimate, which is later refined by multi-view submap constraints in the backend.

3.3 Multi-view Submap Integration

The goal of the backend is to correct the accumulated drift in the global keyframe poses $\{T_i\}$. To achieve this, we periodically construct multi-view submaps that introduce geometrically rich constraints into the global pose graph. Once a sufficient number of consecutive keyframes are accumulated, we construct the m-th submap over a contiguous window of keyframes indexed by $\mathcal{W}_m \subset \{1, \ldots, N\}$. The corresponding keyframe sequence is denoted by $\mathcal{I}_m = \{I_i\}_{i \in \mathcal{W}_m}$. We then apply a feed-forward multi-view model to obtain:

$$\{\hat{D}_{mi}^{\mathrm{mv}}, \hat{T}_{mi}^{\mathrm{mv}}\}_{i \in \mathcal{W}_m} = f_{\mathrm{mv}}(\mathcal{I}_m), \tag{3}$$

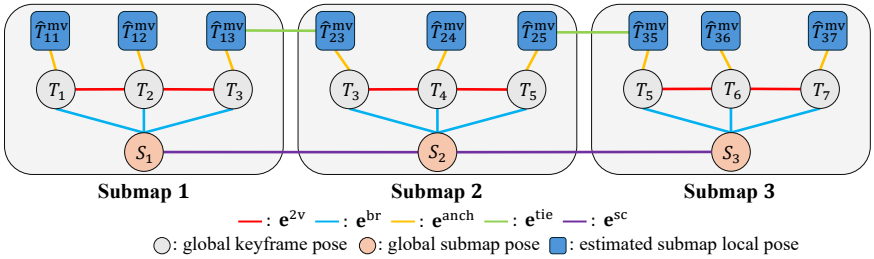

Fig. 3: Illustration of the unified multi-level factor graph.

where $\hat{D}_{mi}^{\text{mv}}$ denotes the predicted depth map for keyframe I_i within the m-th submap, and $\hat{T}_{mi}^{\text{mv}} \in Sim(3)$ represents the submap-local pose of I_i, with its scale component initialized to 1. Although the same geometric foundation model is employed as in the two-view frontend, we denote it by f_{mv} to highlight its operation on multi-view inputs and its asynchronous execution in the backend.

During inference, each submap is reconstructed in its own local coordinate frame, with the center keyframe in $\mathcal{W}_m$ as the submap origin. To integrate this reconstruction into the global trajectory, we introduce a submap pose $S_m \in Sim(3)$ that transforms the m-th submap coordinate frame into the global frame:

$$T_i \approx S_m \hat{T}_{mi}^{\text{mv}}. \tag{4}$$

The optimization therefore jointly estimates the global keyframe poses $\{T_i\}$ and the submap poses $\{S_m\}$. Because multi-view predictions are view-set dependent, the same keyframe can appear with different scales and poses across submaps. Hence, submap-local poses cannot be treated as globally consistent measurements without explicitly introducing $\{S_m\}$. To successfully perform this joint optimization, we initialize the submap pose S_m before joint optimization. Let I_i denote the origin keyframe of submap m, such that $\hat{T}_{mi}^{\text{mv}} = \mathbf{I}$. Let the corresponding global pose obtained from two-view inference be:

$$T_i = \begin{bmatrix} s_i R_i & t_i \\ \mathbf{0}^\top & 1 \end{bmatrix}, \tag{5}$$

where $R_i \in SO(3)$, $t_i \in \mathbb{R}^3$, and $s_i \in \mathbb{R}^+$. To resolve the relative scale ambiguity between two-view and multi-view predictions, we estimate a relative scale using depth statistics $s_{mi}^{\text{rel}} = \text{median}(\hat{D}_{mi}^{\text{mv}} / \hat{D}_i^{\text{2v}})$. Since the multi-view prediction is expressed with unit scale, the initial submap scale is set as $s_m^{(0)} = s_i / s_{mi}^{\text{rel}}$. The submap pose is then initialized in $Sim(3)$ form as:

$$S_m^{(0)} = \begin{bmatrix} s_m^{(0)} R_i & t_i / s_{mi}^{\text{rel}} \\ \mathbf{0}^\top & 1 \end{bmatrix}. \tag{6}$$

This initialization ensures that the submap origin aligns with the global pose of I_i while maintaining a consistent similarity transformation structure. The

translation is scaled by the same relative factor to preserve the similarity transformation between the global and submap frames. It provides a scale-consistent embedding of the submap into the global trajectory, after which both T_i and S_m are jointly refined in the unified optimization.

However, when two-view and multi-view constraints are simultaneously imposed, multiple $Sim(3)$ relationships coexist within a submap. Each keyframe is associated with a global pose T_i, while the submap is associated with its own $Sim(3)$ variable S_m. Consequently, enforcing a single relative scale constraint at initialization is insufficient to guarantee global consistency. This observation motivates the multi-level factor graph formulation, which explicitly models heterogeneous $Sim(3)$ constraints while jointly optimizing $\{T_i\}$ and $\{S_m\}$.

3.4 Unified $Sim(3)$ Pose Graph

As discussed in Sec. 3.3, two-view and multi-view feed-forward inferences produce $Sim(3)$ predictions in different coordinate systems with independent scale. Directly enforcing these heterogeneous constraints in a conventional pose graph may lead to incompatible similarity relations and unstable optimization. To overcome this structural inconsistency, we formulate a unified multi-level pose graph (see Fig. 3) directly on the $Sim(3)$ manifold. Unlike prior pipelines that optimize pairwise or submap constraints in isolation, our formulation jointly models all heterogeneous $Sim(3)$ relations in a single similarity-consistent graph. The unified graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ consists of global pose nodes and submap pose nodes:

$$\mathcal{V} = \mathcal{V}_T \cup \mathcal{V}_S, \quad \mathcal{V}_T = \{T_i \in Sim(3)\}, \quad \mathcal{V}_S = \{S_m \in Sim(3)\}. \tag{7}$$

The edge set is decomposed into three hierarchical groups:

$$\mathcal{E} = \mathcal{E}^{\text{temp}} \cup \mathcal{E}^{\text{v2s}} \cup \mathcal{E}^{\text{s2s}}. \tag{8}$$

These groups encode geometric relations across temporal, cross-level, and inter-submap structures. Specifically, *temporal edges* $\mathcal{E}^{\text{temp}}$ connect consecutive global poses to ensure connectivity. *View-to-submap edges* $\mathcal{E}^{\text{v2s}}$ link global poses T_i with submap poses S_m to align local predictions and constrain relative scales. Finally, *submap-to-submap edges* $\mathcal{E}^{\text{s2s}}$ directly connect overlapping submaps to enforce similarity consistency and prevent scale drift.

View-to-View Edges. The temporal edge set is defined as $\mathcal{E}^{\text{temp}} = \{(i, j) \mid j = i{+}1\}$, where each pair represents a temporally consecutive keyframe pair. For each $(i, j) \in \mathcal{E}^{\text{temp}}$, the two-view feed-forward model provides a relative $Sim(3)$ measurement $\hat{T}_{ij}^{\text{2v}}$. We impose the temporal consistency constraint $T_i^{-1} T_j \approx \hat{T}_{ij}^{\text{2v}}$, which enforces agreement between the composed global poses and the pairwise feed-forward prediction.

Beyond local consistency, these edges form a globally connected temporal backbone. This connectivity is crucial in the unified multi-level graph because,

even when submaps are sparse or non-overlapping, temporal edges allow corrections from higher-level constraints to propagate across the trajectory. The corresponding residual in the Lie algebra $\mathfrak{sim}(3)$ is defined as

$$\mathbf{e}_{ij}^{\mathrm{2v}} = \log\Big((\hat{T}_{ij}^{\mathrm{2v}})^{-1}(T_i^{-1}T_j)\Big). \tag{9}$$

View-to-Submap Edges. The view-to-submap edge set is defined as $\mathcal{E}^{\mathrm{v2s}} = \{(m,i) \mid i \in \mathcal{W}_m\}$, where each element connects a global pose node T_i with its corresponding submap pose node S_m. For each $(m,i) \in \mathcal{E}^{\mathrm{v2s}}$, submap m contains the submap-local pose $\hat{T}_{mi}^{\mathrm{mv}}$. Each view-to-submap edge introduces two complementary residual terms: a pose alignment constraint and a scale consistency constraint.

Pose Alignment. We enforce the $Sim(3)$ consistency relation $T_i \approx S_m\hat{T}_{mi}^{\mathrm{mv}}$, which aligns the submap-local prediction with the global trajectory. The corresponding bridge residual in $\mathfrak{sim}(3)$ is defined as

$$\mathbf{e}_{mi}^{\mathrm{br}} = \log\Big((\hat{T}_{mi}^{\mathrm{mv}})^{-1}S_m^{-1}T_i\Big). \tag{10}$$

Bridge edges align submap-local poses with the global trajectory, correcting drift within $\mathcal{W}_m$. When keyframes are shared across submaps, this coupling implicitly aligns overlapping submaps through their common global pose nodes. However, pose alignment alone does not constrain the relative scale between the global trajectory and the submap coordinate frame.

Scale Anchoring. In addition to pose alignment, each view-to-submap edge also constrains relative scale using per-view depth statistics. Let $\hat{D}_i^{\mathrm{2v}}$ and $\hat{D}_{mi}^{\mathrm{mv}}$ denote the predicted two-view and multi-view depths. The scale residual is defined as

$$\mathbf{e}_{mi}^{\mathrm{anch}} = \log s_i - \log s_m - \log\Big(\mathrm{median}(\hat{D}_{mi}^{\mathrm{mv}}/\hat{D}_i^{\mathrm{2v}})\Big). \tag{11}$$

This additional scale term stabilizes the relative scale between the global trajectory and the submap coordinate frame, preventing scale inconsistency from propagating across submaps.

Submap-to-Submap Edges. While view-to-submap edges enable indirect alignment between submaps via global view nodes, this coupling can be compensated by shifting intermediate view nodes. To enforce strict consistency between submap coordinate frames, we introduce direct submap-to-submap edges for overlapping submaps. The submap-to-submap edge set is defined as $\mathcal{E}^{\mathrm{s2s}} = \{(m,n) \mid \mathcal{W}_m \cap \mathcal{W}_n \neq \emptyset\}$, where an edge is introduced between submaps whose keyframe windows overlap. Let $\mathcal{V}_{mn} = \mathcal{W}_m \cap \mathcal{W}_n$ denote the shared keyframes between submaps m and n. Each submap-to-submap edge introduces two residual terms: a pose consistency constraint (*i.e.*, a tie constraint) and a scale consistency constraint.

Pose Consistency (Tie). For each shared keyframe $i \in \mathcal{V}_{mn}$, multi-view inference yields independent local pose predictions $\hat{T}_{mi}^{\text{mv}}$ and $\hat{T}_{ni}^{\text{mv}}$ within submaps m and n, respectively. Geometric consistency requires that transforming the shared view from both submap coordinate frames yields the same global pose, $S_m \hat{T}_{mi}^{\text{mv}} \approx S_n \hat{T}_{ni}^{\text{mv}}$. We encode this requirement with the tie residual

$$\mathbf{e}_{mn,i}^{\text{tie}} = \log\left((S_m \hat{T}_{mi}^{\text{mv}})^{-1}(S_n \hat{T}_{ni}^{\text{mv}})\right). \tag{12}$$

We also enforce scale consistency between overlapping submaps using dense depth statistics. For each shared keyframe $i \in \mathcal{V}_{mn}$, we estimate a per-view relative scale $\hat{s}_{mn,i} = \text{median}(\hat{D}_{mi}^{\text{mv}}/\hat{D}_{ni}^{\text{mv}})$, where $\hat{D}_{mi}^{\text{mv}}$ and $\hat{D}_{ni}^{\text{mv}}$ denote the multi-view depth predictions for keyframe i obtained within submaps m and n, respectively. We aggregate these estimates across shared views as $\hat{s}_{mn} = \text{median}_{i \in \mathcal{V}_{mn}} \hat{s}_{mn,i}$. The corresponding scale residual is defined as

$$\mathbf{e}_{mn}^{\text{sc}} = \log s_n - \log s_m - \log \hat{s}_{mn}. \tag{13}$$

Using statistics over pixel-aligned depth predictions from shared keyframes provides a robust estimate of the relative scale between submaps. These tie and scale constraints align the coordinate frames and relative scales of overlapping submaps, preventing drift between independently estimated submaps and ensuring that the multi-level graph remains globally consistent. This direct constraint prevents degenerate solutions where submap alignment is satisfied through compensating changes in intermediate view poses.

Optimization. We jointly optimize the global view poses $\{T_i\}$ and submap poses $\{S_m\}$ by minimizing the residuals introduced by the edges in the unified $Sim(3)$ graph. The optimization variables lie on the $Sim(3)$ manifold and are solved using nonlinear least squares. The overall objective is defined as

$$\begin{aligned} \underset{\{T_i\},\{S_m\}}{\arg\min} & \sum_{(i,j)\in\mathcal{E}^{\text{temp}}} \rho\big(\|\mathbf{e}_{ij}^{\text{2v}}\|\big) + \sum_{(m,i)\in\mathcal{E}^{\text{v2s}}} \Big(\rho\big(\|\mathbf{e}_{mi}^{\text{br}}\|\big) + \rho\big(\|\mathbf{e}_{mi}^{\text{anch}}\|\big)\Big) \\ & + \sum_{(m,n)\in\mathcal{E}^{\text{s2s}}} \Big(\sum_{i\in\mathcal{V}_{mn}} \rho\big(\|\mathbf{e}_{mn,i}^{\text{tie}}\|\big) + \rho(\|\mathbf{e}_{mn}^{\text{sc}}\|)\Big), \end{aligned} \tag{14}$$

where $\rho(\cdot)$ denotes the Huber loss function. The optimization is solved using nonlinear least squares with the Levenberg–Marquardt algorithm on the $Sim(3)$ manifold via the Lie algebra $\mathfrak{sim}(3)$. Through this unified optimization, geometric corrections introduced by higher-level submap constraints propagate consistently across the entire trajectory.

Loop Closure. To correct long-term drift, we incorporate a loop closure module following prior works [22]. Loop candidates are detected via global image retrieval and geometric verification between keyframes. Once a loop is detected, we construct a *joint loop submap* that aggregates two multi-view sets centered at the query keyframe and the matched keyframe. The resulting multi-view predictions introduce additional constraints by inserting the loop submap into the unified $Sim(3)$ pose graph. Details are provided in the supplementary material.

Table 1: Quantitative trajectory results on TUM RGB-D. We report ATE RMSE [m]↓ for each sequence under calibrated (Calib.) and uncalibrated (Uncalib.) methods. × denotes failure to produce a valid trajectory. The asterisk on MASt3R-SLAM denotes its uncalibrated setting.

	Method	360	desk	desk2	floor	plant	room	rpy	teddy	xyz	Avg
Calib.	ORB-SLAM3 [3]	×	0.017	0.210	×	0.034	×	×	×	**0.009**	N/A
	DeepV2D [34]	0.243	0.166	0.379	1.653	0.203	0.246	0.105	0.316	0.064	0.375
	DeepFactors [4]	0.159	0.170	0.253	0.169	0.305	0.364	0.043	0.601	0.035	0.233
	DPV-SLAM [19]	0.112	0.018	0.029	0.057	0.021	0.330	0.030	0.084	0.010	0.076
	DPV-SLAM++ [19]	0.132	0.018	0.029	0.050	0.022	0.096	0.032	0.098	0.010	0.054
	GO-SLAM [48]	0.089	**0.016**	0.028	0.025	0.026	0.052	**0.019**	0.048	0.010	0.035
	DROID-SLAM [35]	0.111	0.018	0.042	**0.021**	**0.016**	**0.049**	0.026	0.048	0.012	0.038
	MASt3R-SLAM [26]	**0.049**	**0.016**	**0.024**	0.025	0.020	0.061	0.027	**0.041**	**0.009**	**0.030**
Uncalib.	CUT3R [39]	0.174	0.592	0.546	0.662	0.467	0.911	0.051	0.845	0.129	0.486
	SLAM3R [21]	0.211	0.861	0.967	0.790	0.755	1.013	0.063	0.986	0.185	0.648
	MASt3R-SLAM* [26]	0.070	0.032	0.055	0.056	0.035	0.118	0.041	0.116	0.020	0.060
	VGGT-SLAM [22]	**0.063**	0.031	0.048	0.152	**0.023**	0.133	0.038	0.039	0.020	0.061
	ViSTA-SLAM [46]	0.104	0.030	0.030	0.070	0.052	0.067	0.023	0.080	0.015	0.052
	`UniSim-SLAM`	0.067	**0.018**	**0.022**	**0.034**	0.029	**0.056**	**0.021**	**0.031**	**0.013**	**0.032**

4 Experiment

4.1 Experimental Setup

We evaluate `UniSim-SLAM` on two standard RGB SLAM benchmarks: TUM RGB-D [32] and 7-Scenes [30]. For 7-Scenes, we utilize the refined ground-truth poses provided by Brachmann *et al.* [2]. To evaluate tracking performance, we measure the Root Mean Square Error (RMSE) of the Absolute Trajectory Error (ATE) following $Sim(3)$ alignment, computed via the evo toolkit [10]. Furthermore, to assess dense 3D reconstruction quality, we report Accuracy, Completion, and Chamfer Distance on the 7-Scenes dataset.

We compare `UniSim-SLAM` against recent learning-based SLAM systems, including two-view feed-forward approaches (MASt3R-SLAM [26], ViSTA-SLAM [46]), multi-view approaches (VGGT-SLAM [22]), and regression-based methods (CUT3R [39], SLAM3R [21]). Calibration-based methods [3, 4, 19, 34, 48] assuming known intrinsics are also included. For TUM RGB-D, we report baseline results as reported in prior works [22, 26, 46].

Implementation Details. For 7-Scenes, we use a keyframe selection stride of 5 for all uncalibrated methods. For TUM RGB-D, we follow the evaluation protocol of [46] and use a stride of 3. For backend optimization, `UniSim-SLAM` constructs submaps of size 16 with an overlap of 2 frames. Additional implementation details, frontend and runtime analyses, large-scale evaluations, and extended reconstruction results are provided in the supplementary material.

4.2 Evaluation

Camera Trajectory. `UniSim-SLAM` achieves state-of-the-art performance in the uncalibrated setting on both TUM RGB-D (Tab. 1) and 7-Scenes (Tab. 2). On TUM RGB-D, we observe a noticeable reduction in trajectory error on the

Table 2: Quantitative trajectory results on 7-Scenes (ATE RMSE [m]↓).

	Method	chess	fire	heads	office	pumpkin	kitchen	stairs	Avg
Calib.	DROID-SLAM [35]	**0.018**	**0.027**	**0.021**	**0.041**	**0.025**	**0.016**	**0.017**	**0.024**
	MASt3R-SLAM [26]	0.082	0.030	0.024	0.052	0.050	0.044	0.027	0.044
Uncalib.	CUT3R [39]	0.514	0.110	0.197	0.430	0.346	0.202	0.385	0.312
	SLAM3R [21]	0.131	0.044	0.040	0.058	0.100	0.064	0.116	0.079
	MASt3R-SLAM [26]	0.090	0.058	0.039	0.072	0.084	0.062	0.071	0.068
	VGGT-SLAM [22]	0.039	0.024	0.041	0.032	0.050	0.034	0.042	0.037
	ViSTA-SLAM [46]	0.075	0.035	0.030	0.064	0.065	0.041	0.036	0.049
	`UniSim-SLAM`	**0.017**	**0.018**	**0.026**	**0.024**	**0.022**	**0.016**	**0.019**	**0.020**

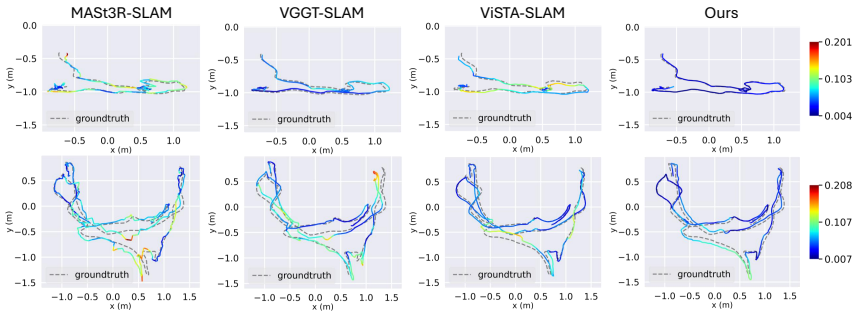

Fig. 4: Qualitative trajectory comparisons on 7-Scenes [30].

floor sequence. Because this scene is dominated by planar structures, it provides limited geometric cues for scale recovery, making feed-forward predictions prone to scale drift. Our method significantly improves trajectory accuracy in this challenging case, indicating that the proposed multi-level factor graph effectively stabilizes scale by jointly optimizing view and submap constraints (see Fig. 4).

A similar trend is observed on 7-Scenes. In the *chess* sequence, large depth variations and changes in camera-to-object distance introduce scale inconsistencies, highlighting the sensitivity of feed-forward models to input view configurations. `UniSim-SLAM` substantially reduces trajectory error on this sequence, demonstrating that unified $Sim(3)$ optimization effectively mitigates scale drift.

3D Reconstruction. To evaluate 3D reconstruction performance, we reconstruct a global point cloud using the optimized global view poses $\{T_i\}$ together with the multi-view depth estimates $\hat{D}^{\text{mv}}$ and intrinsics, and confidence maps. Following [22], we discard points corresponding to the lowest 25% confidence scores. Because our framework produces overlapping submaps, directly aggregating all reconstructed points would artificially improve completion metrics due to increased point density. To avoid this bias, we select a single depth map with the highest confidence score for each view from the submaps.

Tab. 3 shows that `UniSim-SLAM` achieves lower Accuracy and Chamfer Distance while maintaining comparable Completion, indicating improved geometric

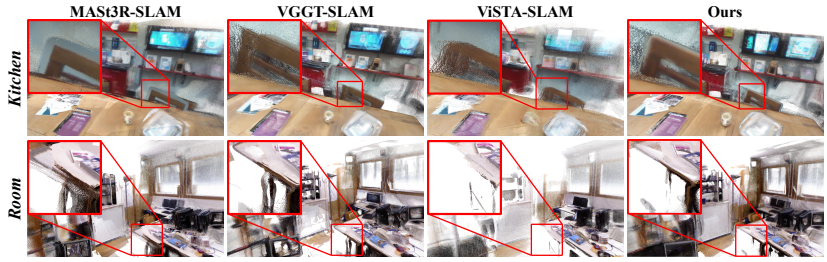

Fig. 5: Reconstruction results on 7-Scenes and TUM RGB-D. Red boxes indicate close-up views highlighting scene details.

precision of the reconstructed scene. Fig. 5 presents qualitative results on the 7-Scenes *kitchen* and TUM RGB-D *room* sequences.

Latency. Tracking latency is an important aspect of SLAM systems, as camera pose estimates are expected to be available with low latency once input frames arrive. We therefore analyze the latency required to obtain a pose estimate during tracking. Specifically, we measure the elapsed time from receiving the minimum required input frames to producing a camera pose estimate in the frontend, and report the results in Tab. 4c. We observe that `UniSim-SLAM` achieves lower ATE by using multi-view information while operating at a moderate latency. This suggests that the proposed system alleviates the accuracy-efficiency trade-off, although the latency remains higher than that of specialized two-view SLAM pipelines. This difference mainly arises from the large model size and general-purpose nature of our frontend backbone, VGGT. To further analyze this aspect, we additionally evaluate a variant that replaces the frontend with a low-latency model (Ours + STA in Tab. 4c). Although using different models in the frontend and backend introduces additional $Sim(3)$ inconsistencies, the resulting system still achieves better performance than existing methods. These results indicate that our multi-level factor graph can effectively perform $Sim(3)$ refinement even when integrating predictions from heterogeneous feed-forward models.

4.3 Ablation Study

In this section, we analyze the impact of the submap configuration and the contribution of each graph constraint. Additional ablation studies and extended experimental results are provided in the supplementary material.

Impact of Submap Configuration. In Tab. 4b, we analyze two key hyperparameters: the submap size w, which determines the number of keyframes per submap, and the submap overlap ϕ, which controls the number of shared keyframes between adjacent submaps.

Table 3: Reconstruction error on 7-Scenes. We report Accuracy (Acc.), Completion (Comp.), and Chamfer Distance.

	Method	Acc. ↓	Comp. ↓	Chamfer ↓
Calib.	DROID-SLAM [35]	0.111	**0.049**	0.080
	MASt3R-SLAM [26]	**0.064**	0.068	**0.066**
Uncalib.	Spann3R @5 [37]	0.095	**0.041**	0.068
	CUT3R [39]	0.107	0.059	0.083
	SLAM3R [21]	0.069	0.153	0.111
	MASt3R-SLAM [26]	0.054	0.048	0.051
	VGGT-SLAM [22]	0.039	0.051	0.045
	ViSTA-SLAM [46]	0.041	0.056	0.049
	`UniSim-SLAM`	**0.035**	0.046	**0.041**

Table 4: Ablation studies and latency comparison.

(a) Ablation Studies on 7-Scenes.

	w/o Backend	w/o LC	w/o $\mathbf{e}^{2v}$	w/o $\mathbf{e}^{anch}$	w/o $\mathbf{e}^{br}$	w/o $\mathbf{e}^{tie}$	w/o $\mathbf{e}^{sc}$	Ours (full)
$\phi = 0$	0.124	0.061	0.101	0.083	0.116	0.040	0.048	0.032
$\phi = 2$	0.124	0.037	0.021	0.020	0.063	0.027	0.031	0.020

(b) w and ϕ on TUM RGB-D.

w	$\phi = 1$	$\phi = 2$	$\phi = 4$	$\phi = 8$
4	0.043	0.041	–	–
8	0.035	0.035	0.035	–
16	0.031	0.032	0.031	0.030
32	0.033	0.032	0.034	0.033

(c) Latency comparison on 7-Scenes.

Method	Frontend	Latency [ms]↓	ATE↓
MASt3R-SLAM [26]	MASt3R [17]	90	0.068
VGGT-SLAM [22]	VGGT [38]	3410	0.037
ViSTA-SLAM [46]	STA [46]	**35**	0.049
Ours + STA [46]	STA [46]	**35**	0.027
Ours	VGGT [38]	197	**0.020**

Submap Overlap. As shown in Tab. 4b, an overlap of a single frame ($\phi = 1$) makes submap alignment rely entirely on a single multi-view local pose estimation, risking optimization instability. To prevent this, we adopt $\phi = 2$ as the default configuration, ensuring robust optimization with minimal overhead.

Submap Size. Larger submaps improve local multi-view estimation but reduce the number of submap nodes in the graph, weakening global constraints. Accordingly, large submaps ($w = 32$) do not necessarily improve performance. We thus set $w = 16$ by default. Additional experiments and details are in the supplementary material.

Effectiveness of Graph Constraints. In Tab 4a, we provide an ablation study evaluating the contribution of each backend component. Under the default configuration ($\phi = 2$), removing any individual constraint leads to a noticeable performance drop, indicating that the proposed multi-level factor graph effectively refines the global trajectory. This confirms that the improvements

arise from the synergistic interaction of multiple constraints rather than a single dominant factor. Notably, even without loop closure, the system achieves performance comparable to existing state-of-the-art methods, suggesting that the proposed graph alone produces a globally consistent optimization. Furthermore, in a non-overlap setting ($\phi = 0$), the temporal view-to-view edge becomes particularly crucial, as it maintains graph connectivity and enables scale corrections to propagate across the trajectory.

5 Conclusion

We proposed `UniSim-SLAM`, a feed-forward SLAM system that unifies low-latency two-view tracking and multi-view submaps within a $Sim(3)$ optimization framework. Our key insight is that predictions from different feed-forward inference regimes reside in heterogeneous coordinate systems and must therefore be jointly optimized. By representing these predictions as complementary constraints in a multi-level factor graph, `UniSim-SLAM` enables consistent global pose estimation while preserving immediate tracking updates. This unified formulation bridges fast two-view inference and constraint-rich multi-view reconstruction, enabling robust drift correction. Experiments on TUM RGB-D and 7-Scenes demonstrate that `UniSim-SLAM` outperforms existing feed-forward SLAM methods, highlighting the effectiveness of unified optimization for integrating heterogeneous geometric predictions.

Acknowledgements

This work was supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.RS-2020-II201336, Artificial Intelligence Graduate School Program (UNIST); No.RS-2022-II220907, Development of AI Bots Collaboration Platform and Self-organizing; No.RS-2026-25507551, Development of Egocentric Data Sensing and Spatial Immersive Experience Technology), and by the InnoCORE program of the Ministry of Science and ICT (25-InnoCORE-01).

References

1. Agarwal, S., Furukawa, Y., Snavely, N., Simon, I., Curless, B., Seitz, S.M., Szeliski, R.: Building Rome in a day. Communications of the ACM **54**(10), 105–112 (2011)
2. Brachmann, E., Humenberger, M., Rother, C., Sattler, T.: On the limits of pseudo ground truth in visual camera re-localisation. In: Int. Conf. Comput. Vis. pp. 6198–6208 (2021). `https://doi.org/10.1109/ICCV48922.2021.00616`
3. Campos, C., Elvira, R., Rodríguez, J.J.G., Montiel, J.M.M., Tardós, J.D.: ORB-SLAM3: An accurate open-source library for visual, visual–inertial, and multimap SLAM. IEEE Trans. Robot. **37**(6), 1874–1890 (2021). `https://doi.org/10.1109/TRO.2021.3075644`

4. Czarnowski, J., Laidlow, T., Clark, R., Davison, A.J.: DeepFactors: Real-time probabilistic dense monocular SLAM. IEEE Robot. Autom. Lett. **5**(2), 721–728 (2020). https://doi.org/10.1109/LRA.2020.2965415
5. Deng, K., Ti, Z., Xu, J., Yang, J., Xie, J.: VGGT-Long: Chunk it, loop it, align it–pushing VGGT's limits on kilometer-scale long RGB sequences. arXiv preprint arXiv:2507.16443 (2025)
6. Dong, S., Wang, S., Liu, S., Cai, L., Fan, Q., Kannala, J., Yang, Y.: Reloc3r: Large-scale training of relative camera pose regression for generalizable, fast, and accurate visual localization. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 16739–16752 (2025). https://doi.org/10.1109/CVPR52734.2025.01560
7. Duisterhof, B.P., Zust, L., Weinzaepfel, P., Leroy, V., Cabon, Y., Revaud, J.: MASt3R-SfM: A fully-integrated solution for unconstrained structure-from-motion. In: Int. Conf. 3D Vis. pp. 1–10 (2025). https://doi.org/10.1109/3DV66043.2025.00008
8. Engel, J., Koltun, V., Cremers, D.: Direct sparse odometry. IEEE Trans. Pattern Anal. Mach. Intell. **40**(3), 611–625 (2018). https://doi.org/10.1109/TPAMI.2017.2658577
9. Engel, J., Schöps, T., Cremers, D.: LSD-SLAM: Large-scale direct monocular SLAM. In: Eur. Conf. Comput. Vis. pp. 834–849. Springer (2014)
10. Grupp, M.: evo: Python package for the evaluation of odometry and SLAM. https://github.com/MichaelGrupp/evo (2017)
11. Kaess, M., Ranganathan, A., Dellaert, F.: iSAM: Incremental smoothing and mapping. IEEE Trans. Robot. **24**(6), 1365–1378 (2008)
12. Karaev, N., Rocco, I., Graham, B., Neverova, N., Vedaldi, A., Rupprecht, C.: CoTracker: It is better to track together. In: Eur. Conf. Comput. Vis. pp. 18–35. Springer (2024)
13. Keetha, N., Müller, N., Schönberger, J., Porzi, L., Zhang, Y., Fischer, T., Knapitsch, A., Zauss, D., Weber, E., Antunes, N., et al.: MapAnything: Universal feed-forward metric 3D reconstruction. In: Int. Conf. 3D Vis. pp. 499–509 (2026)
14. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G.: 3D gaussian splatting for real-time radiance field rendering. ACM Trans. Graph. **42**(4) (Jul 2023)
15. Kneip, L., Scaramuzza, D., Siegwart, R.: A novel parametrization of the perspective-three-point problem for a direct computation of absolute camera position and orientation. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 2969–2976 (2011). https://doi.org/10.1109/CVPR.2011.5995464
16. Lepetit, V., Moreno-Noguer, F., Fua, P.: EPnP: An accurate O(n) solution to the PnP problem. Int. J. Comput. Vis. **81**(2), 155–166 (2009)
17. Leroy, V., Cabon, Y., Revaud, J.: Grounding image matching in 3D with MASt3R. In: Eur. Conf. Comput. Vis. pp. 71–91. Springer (2024)
18. Lin, H., Chen, S., Liew, J.H., Chen, D.Y., Li, Z., Zhao, Y., Peng, S., Guo, H., Zhou, X., Shi, G., Feng, J., Kang, B.: Depth Anything 3: Recovering the visual space from any views. In: Int. Conf. Learn. Represent. (2026), https://openreview.net/forum?id=yirunib8l8
19. Lipson, L., Teed, Z., Deng, J.: Deep patch visual SLAM. In: Eur. Conf. Comput. Vis. pp. 424–440. Springer (2024). https://doi.org/10.1007/978-3-031-72627-9_24
20. Liu, S., Gao, Y., Zhang, T., Pautrat, R., Schönberger, J.L., Larsson, V., Pollefeys, M.: Robust incremental structure-from-motion with hybrid features. In: Eur. Conf. Comput. Vis. pp. 249–269. Springer (2024)

21. 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: IEEE Conf. Comput. Vis. Pattern Recog. pp. 16651–16662 (2025). https://doi.org/10.1109/CVPR52734.2025.01552
22. Maggio, D., Lim, H., Carlone, L.: VGGT-SLAM: Dense RGB SLAM optimized on the SL(4) manifold. Adv. Neural Inform. Process. Syst. **38** (2025)
23. Matsuki, H., Murai, R., Kelly, P.H.J., Davison, A.J.: Gaussian splatting SLAM. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 18039–18048 (2024). https://doi.org/10.1109/CVPR52733.2024.01708
24. Mur-Artal, R., Montiel, J.M.M., Tardós, J.D.: ORB-SLAM: A versatile and accurate monocular SLAM system. IEEE Trans. Robot. **31**(5), 1147–1163 (2015). https://doi.org/10.1109/TRO.2015.2463671
25. Mur-Artal, R., Tardós, J.D.: ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras. IEEE Trans. Robot. **33**(5), 1255–1262 (2017). https://doi.org/10.1109/TRO.2017.2705103
26. Murai, R., Dexheimer, E., Davison, A.J.: MASt3R-SLAM: Real-time dense SLAM with 3D reconstruction priors. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 16695–16705 (2025). https://doi.org/10.1109/CVPR52734.2025.01556
27. Oquab, M., Darcet, T., Moutakanni, T., Vo, H.V., Szafraniec, M., Khalidov, V., Fernandez, P., HAZIZA, D., Massa, F., El-Nouby, A., Assran, M., Ballas, N., Galuba, W., Howes, R., Huang, P.Y., Li, S.W., Misra, I., Rabbat, M., Sharma, V., Synnaeve, G., Xu, H., Jegou, H., Mairal, J., Labatut, P., Joulin, A., Bojanowski, P.: DINOv2: Learning robust visual features without supervision. Trans. Mach. Learn Res. (2024), https://openreview.net/forum?id=a68SUt6zFt
28. Rosinol, A., Abate, M., Chang, Y., Carlone, L.: Kimera: An open-source library for real-time metric-semantic localization and mapping. In: IEEE Int. Conf. Robot. Autom. pp. 1689–1696 (2020). https://doi.org/10.1109/ICRA40945.2020.9196885
29. Schönberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 4104–4113 (2016). https://doi.org/10.1109/CVPR.2016.445
30. Shotton, J., Glocker, B., Zach, C., Izadi, S., Criminisi, A., Fitzgibbon, A.: Scene coordinate regression forests for camera relocalization in RGB-D images. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 2930–2937 (2013). https://doi.org/10.1109/CVPR.2013.377
31. Smart, B., Zheng, C., Laina, I., Prisacariu, V.A.: Splatt3r: Zero-shot gaussian splatting from uncalibrated image pairs. arXiv preprint arXiv:2408.13912 (2024)
32. Sturm, J., Engelhard, N., Endres, F., Burgard, W., Cremers, D.: A benchmark for the evaluation of RGB-D SLAM systems. In: IEEE/RSJ Int. Conf. Intell. Robots Syst. pp. 573–580 (2012). https://doi.org/10.1109/IROS.2012.6385773
33. Sucar, E., Liu, S., Ortiz, J., Davison, A.J.: iMAP: Implicit mapping and positioning in real-time. In: Int. Conf. Comput. Vis. pp. 6209–6218 (2021). https://doi.org/10.1109/ICCV48922.2021.00617
34. Teed, Z., Deng, J.: DeepV2D: Video to depth with differentiable structure from motion. In: Int. Conf. Learn. Represent. (2020)
35. Teed, Z., Deng, J.: DROID-SLAM: Deep visual SLAM for monocular, stereo, and RGB-D cameras. In: Adv. Neural Inform. Process. Syst. vol. 34, pp. 16558–16569 (2021)
36. Triggs, B., McLauchlan, P., Hartley, R., Fitzgibbon, A.: Bundle adjustment—a modern synthesis. Vision Algorithms: Theory and Practice pp. 298–372 (2000)

37. Wang, H., Agapito, L.: 3D reconstruction with spatial memory. In: Int. Conf. 3D Vis. pp. 78–89 (2025). https://doi.org/10.1109/3DV66043.2025.00013
38. Wang, J., Chen, M., Karaev, N., Vedaldi, A., Rupprecht, C., Novotny, D.: VGGT: Visual geometry grounded transformer. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 5294–5306 (2025). https://doi.org/10.1109/CVPR52734.2025.00499
39. Wang, Q., Zhang, Y., Holynski, A., Efros, A.A., Kanazawa, A.: Continuous 3D perception model with persistent state. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 10510–10522 (2025). https://doi.org/10.1109/CVPR52734.2025.00983
40. Wang, S., Leroy, V., Cabon, Y., Chidlovskii, B., Revaud, J.: DUSt3R: Geometric 3D vision made easy. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 20697–20709 (2024). https://doi.org/10.1109/CVPR52733.2024.01956
41. Wang, Y., Zhou, J., Zhu, H., Chang, W., Zhou, Y., Li, Z., Chen, J., Pang, J., Shen, C., He, T.: π^3: Permutation-equivariant visual geometry learning. In: Int. Conf. Learn. Represent. (2026), https://openreview.net/forum?id=DTQIjngDta
42. Wen, B., Trepte, M., Aribido, J., Kautz, J., Gallo, O., Birchfield, S.: FoundationStereo: Zero-shot stereo matching. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 5249–5260 (2025). https://doi.org/10.1109/CVPR52734.2025.00495
43. Yan, C., Qu, D., Xu, D., Zhao, B., Wang, Z., Wang, D., Li, X.: GS-SLAM: Dense visual SLAM with 3D gaussian splatting. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 19595–19604 (2024). https://doi.org/10.1109/CVPR52733.2024.01853
44. Yang, L., Kang, B., Huang, Z., Xu, X., Feng, J., Zhao, H.: Depth Anything: Unleashing the power of large-scale unlabeled data. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 10371–10381 (2024). https://doi.org/10.1109/CVPR52733.2024.00987
45. Yang, L., Kang, B., Huang, Z., Zhao, Z., Xu, X., Feng, J., Zhao, H.: Depth Anything V2. Adv. Neural Inform. Process. Syst. **37**, 21875–21911 (2024)
46. Zhang, G., Qian, S., Wang, X., Cremers, D.: ViSTA-SLAM: Visual SLAM with symmetric two-view association. In: Int. Conf. 3D Vis. pp. 396–406 (2026). https://doi.org/10.1109/3DV69130.2026.00044
47. Zhang, G., Sandström, E., Zhang, Y., Patel, M., Van Gool, L., Oswald, M.R.: GlORIE-SLAM: Globally optimized RGB-only implicit encoding point cloud SLAM. arXiv preprint arXiv:2403.19549 (2024)
48. Zhang, Y., Tosi, F., Mattoccia, S., Poggi, M.: GO-SLAM: Global optimization for consistent 3D instant reconstruction. In: Int. Conf. Comput. Vis. pp. 3704–3714 (2023). https://doi.org/10.1109/ICCV51070.2023.00345
49. Zhu, Z., Peng, S., Larsson, V., Cui, Z., Oswald, M.R., Geiger, A., Pollefeys, M.: NICER-SLAM: Neural implicit scene encoding for RGB SLAM. In: Int. Conf. 3D Vis. pp. 42–52 (2024). https://doi.org/10.1109/3DV62453.2024.00096