S^3: SEMANTIC SCALE SYNTHESIS FOR TRAINING-FREE METRIC DEPTH FROM COMPLEMENTARY FOUNDATION MODELS IN MONOCULAR 3DGS SLAM

Anonymous authors
Paper under double-blind review

ABSTRACT

Monocular depth foundation models can predict metric depth from individual images, but these predictions vary in scale and do not provide the multi-view geometric consistency required for estimation of motion while building a map. We introduce S^3-SLAM, a training-free monocular 3D Gaussian Splatting SLAM system that synthesizes two complementary foundation models: UniDepthV2, which infers single-frame metric depth, and VGGT, which infers multi-view-consistent but scale-free geometry. We show that their depth errors are complementary across spatial depth frequencies, introducing an online method of frequency-based fusion that exploits this structure while decoupling the mapping and tracking steps of SLAM. We use persistent semantic object landmarks to further correct frame-to-frame drift on short timescales, and to provide long-term geometric consistency when revisiting previously observed regions. By evaluating this approach across a range of indoor SLAM benchmarks, we demonstrate recovery of consistent metric-scale depth and trajectory estimates.

1 INTRODUCTION

Monocular depth estimation aims to infer 3D scene structure from an individual image, but for a robot, this geometry must remain consistent across multiple sequential observations as the camera moves through the environment. Maintaining that consistency requires jointly reasoning about both the robot's motion and the structure of its environment. Simultaneous Localization and Mapping (SLAM) (Leonard & Durrant-Whyte, 1991) addresses this challenge by estimating camera motion while *simultaneously* building a map of its surroundings. This process is critical because it enables a robot to maintain a spatially consistent map of an unknown environment while simultaneously tracking its own motion, making it fundamental for autonomous robot navigation (Weiss et al., 2011). The resulting pose estimates and maps support downstream tasks including navigation, trajectory optimization (Lim et al., 2026) and collision avoidance (Tscholl et al., 2026). Without metric scale, the map can be geometrically consistent yet fail to tell the robot how far away an obstacle actually is.

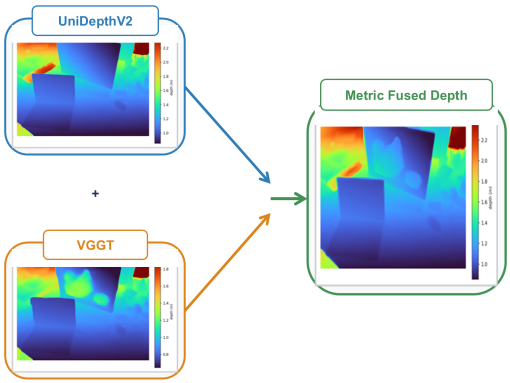

Figure 1: **Complementary Depth Fusion.** Fusing two complementary foundation models yields geometrically-consistent metric depth.

Recent monocular depth foundation models (DFMs) provide sources of metric information by predicting metric depth maps from single images (Piccinelli et al., 2026). However, directly incorporating these maps into SLAM is not straightforward due to variation in scale or structure across subsequent observations. SLAM additionally requires geometric consistency over a continuous camera trajectory, due to how observations from different viewpoints must be integrated into the same

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

map. While the Visual Geometry Grounded Transformer (VGGT) model (Wang et al., 2025) provides multi-view-consistent geometry and correspondence and has been used previously in SLAM (Maggio et al., 2025), it does not provide absolute metric scale. This motivates the problem of online synthesis of single-frame metric-level DFMs with multi-view-consistent, scale-free geometry foundation models (GFMs).

The distinction between scale-free and metric-level depth information becomes particularly important in vision-based SLAM systems that use 3D Gaussian Splatting (3DGS) (Kerbl et al., 2023; Tosi et al., 2026). 3DGS uses compact Gaussian primitives to explicitly represent a scene while providing a continuous and differentiable map representation (Kerbl et al., 2023). In differentiable 3DGS SLAM (Keetha et al., 2024), Gaussians and camera poses can be optimized online through rendering losses. As a result, errors in depth supervision not only affect an individual frame, but can also influence persistent map geometry and subsequent pose estimation during trajectory tracking. Existing monocular 3DGS SLAM approaches therefore commonly rely on photometric and relative geometric consistency (Matsuki et al., 2024; Yu et al., 2025; Cheng et al., 2025), rather than directly maintaining metrically consistent depth throughout mapping.

We address this challenge with 3DGS SLAM by decoupling how the mapping and pose estimation components of SLAM consume depth information. The mapping component fuses metric depth from UniDepthV2 with geometrically consistent depth from VGGT through an online optimization process that estimates reliability of the two depth sources. Simultaneously, the tracking component uses a scale-free version of this same depth map, enabling pose estimation to remain geometrically consistent while allowing the map to retain metric scale.

Another mechanism by which we improve geometric consistency in our map is through semantic object landmarks, which are identified through Segment Anything Model 2 (SAM2) (Ravi et al., 2024). Prior work has used semantic landmarks to improve both feature correspondence (Gan et al., 2023) and map consistency (Kim & Kim, 2025), as the identity of an object remains stable across changes in viewpoint. Once these landmarks are identified, we accumulate observations of them within the map over frames, which we can then compare against the current depth map, and use to further improve consistency via drift correction.

Integrating these mechanisms together in a semantic monocular RGB 3DGS SLAM system, S^3-*SLAM*, our contributions are fourfold:

- We present an online, training-free fusion of two complementary monocular foundation models: VGGT's multi-view-consistent but scale-free geometry and UniDepthV2's metric single-frame depth.
- We decouple the mapping and tracking, and introduce a method of online reliability optimization through spatial depth frequency analysis.
- We show that persistent semantic object landmarks provide an additional online signal for correcting frame-dependent metric-depth scale error, while also supporting long-term drift correction.
- We evaluate metric-level localization and mapping performance across TUM RGB-D (Sturm et al., 2012), 7-Scenes (Glocker et al., 2013), and ScanNet (Dai et al., 2017).

2 Related Work

Scene Representations: Dense scene representations encode geometry, color, and appearance throughout the observed environment. Neural Radiance Fields (Mildenhall et al., 2021), a neural-implicit method, use neural networks to predict density and color along a ray, but this approach suffers from a high computational cost due to the large number of MLP evaluations needed (Huang et al., 2024). 3DGS mitigates this issue through its explicit representation and avoidance of repeated neural-field evaluations at rendering time (Tosi et al., 2026; Zhang et al., 2023; Yan et al., 2024). Our method uses 3DGS because it is a method that can be updated online while directly incorporating and optimizing depth.

Monocular SLAM: Monocular SLAM estimates camera motion and map structure from a single RGB camera, causing metric scale to be ambiguous without additional information (Davison et al.,

2007). The ORB-SLAM algorithm, introduced by Mur-Artal et al. (2015) is a well-known feature-based algorithm which uses Oriented FAST and Rotated BRIEF (ORB) features (Rublee et al., 2011) as part of the optimization process, obtaining real-time performance without GPUs. While recent monocular 3DGS algorithms (Matsuki et al., 2024; Zhang et al., 2025) primarily optimize relative geometric and photometric consistency, and others (Li et al., 2026) use reconstruction priors for geometric consistency, our approach focuses on maintaining metric-depth consistency during online optimization.

Semantic SLAM: Semantic SLAM utilizes object or category-level information in localization and mapping, rather than treating segmentation as a separate perception output (Pu et al., 2023). Prior work has used semantic information to improve pose estimation, map consistency, and object-level mapping (Gan et al., 2017; Zhang et al., 2018). Our method instead uses semantics to provide persistent object-level references for the correction of temporal metric-depth variation.

3 Background

3.1 3D Gaussian Splatting

Introduced by Kerbl et al. (2023), 3D Gaussian Splatting (3DGS) takes a set of images of a static scene, with corresponding cameras calibrated by Structure from Motion (SfM), and optimizes a set of Gaussian primitives to represent the scene. We follow several of the 3DGS methods used by Keetha et al. (2024) when integrated into a SLAM pipeline: namely the silhouette-based optimization and the isotropic Gaussian formulation with view-independent color, where each Gaussian is parameterized by a 3D center, scalar radius, color, and opacity. The key equations and components of this method are given in Appendix A.1.1.

3.2 Visual Geometry Grounded Transformer (VGGT)

VGGT, introduced by Wang et al. (2025), is a large feed-forward transformer that takes in a series of images as an input with the first one assumed to be the base reference frame (more formalism in A.2). It predicts camera parameters, depth maps, point maps, and a grid of features. The translation and depth estimates are represented in a shared scene coordinate system without metric scale, following Eq. (1)

$$d_{\text{im}} = d_{\text{vggt}} \cdot \gamma_{\text{vggt}},\ m_{\text{im}} = m_{\text{vggt}} \cdot \gamma_{\text{vggt}}. \tag{1}$$

Here, d is depth, m is translation, and γ_{vggt} is the unknown scale factor mapping VGGT's internal scene scale to metric units; this scale factor is not output during inference. Thus, VGGT provides multi-view-consistent depth and translation estimates, while leaving their shared metric scale undetermined.

3.3 Factor Graph Optimization

Factor graphs have been widely used in robotics and perception for SLAM and estimation (Dellaert & Kaess, 2017). We lay out our trajectory estimate as a nonlinear factor graph over camera poses $X = \{x_0, \ldots, x_t\}$, where each x_t is a camera pose in $SE(3)$, and solve it incrementally using incremental Smoothing and Mapping (iSAM2) (Kaess et al., 2011), allowing us to fuse incoming pose estimates from VGGT in an online manner. We provide further background on factor graphs and iSAM2 in Appendix A.3.

4 Monocular SLAM Architecture

4.1 SLAM Terminology

We define several SLAM-specific terms that will be used in this section. First, we define a *keyframe* as a subset of frames where computationally-expensive operations (such as VGGT inference) are called. While several classical SLAM algorithms (Mur-Artal et al., 2015) determine them based on parallax or covisibility, we selected them at a fixed cadence (i.e. every n-th frame). *Loop closure* is a mechanism by which drift in pose estimation is corrected by identifying when a robot has reached

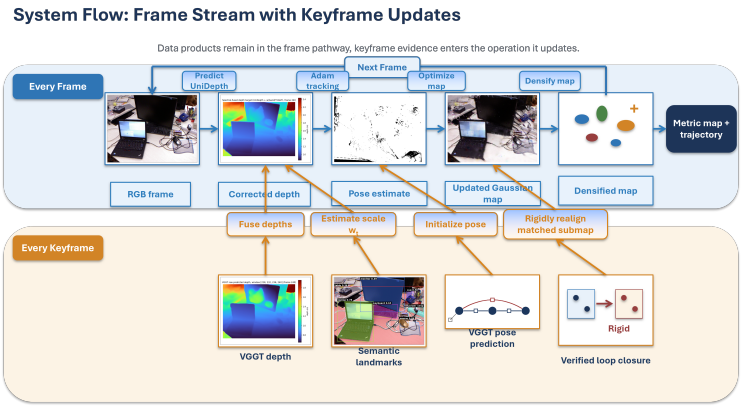

Figure 2: **Algorithm Overview.** The architecture of our algorithm is broken down into four distinct parts: depth fusion, tracking, loop closure detection, and Gaussian densification.

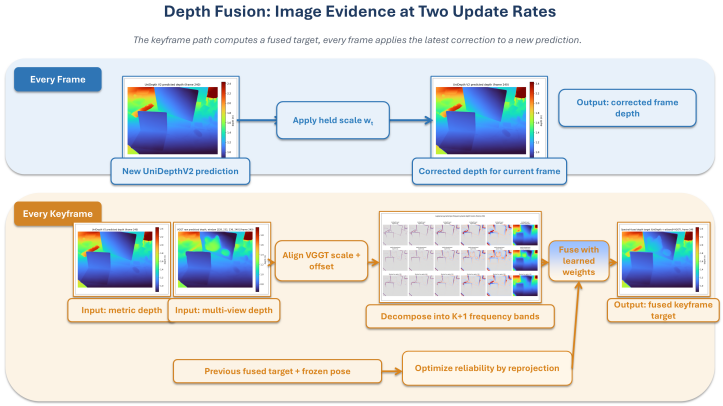

Figure 3: **Depth Fusion.** Our algorithm breaks down the spatial depth frequencies of VGGT and UniDepthV2 to produce a fused depth target for the Gaussian Splatting optimizer.

a previously visited point (Hess et al., 2016). Several strategies can be used for correcting this drift, including the breaking down of a global map into locally constructed chunks called *submaps* which are then re-aligned (Maggio & Carlone, 2026; Zhu et al., 2025). The *tracking* component of SLAM refers to the estimation of the camera pose of incoming frames relative to the existing map and previous observations.

4.2 Algorithm Architecture Overview

We adopt the isotropic Gaussian and silhouette-based optimization of the RGB-D 3DGS SLAM system SplaTAM (Keetha et al., 2024) (given in Appendix A.1.1), but introduce four methods to maintain metric and geometric consistency over long monocular trajectories. They address four key estimation challenges: (1) combining single-view metric and multi-view-consistent scale-free predictions, (2) correcting frame-dependent variation from foundation model predictions, (3) recovering geometric consistency via loop closures, and (4) using VGGT to anchor pose estimates. A flowchart of the full system is shown in Fig. 2.

4.3 Online Depth Fusion Via Spatial Depth Frequencies

For each incoming frame, UniDepthV2 provides a single-frame metric-level depth estimate that can be used by the SLAM system; see Appendix A.1.2. At every keyframe kf_t, we additionally run VGGT inference over a sliding window of the most recent keyframes up to kf_{t-1}, producing multi-

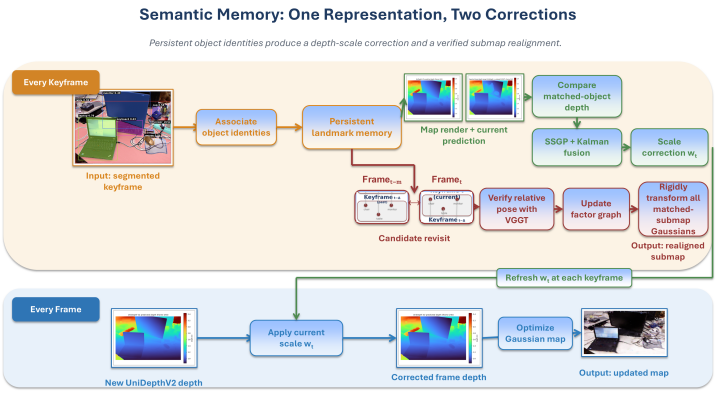

Figure 4: **Semantic Analysis Overview.** The semantics are used in two ways: to provide more robust depth estimation, and for loop closure/rigid submap corrections

view, scale-free depth estimates. We then perform depth fusion, which consists of three sequential components: (1) metric alignment of VGGT, (2) computation of cross-model reliability, and (3) frequency-dependent depth fusion.

VGGT infers multi-view-consistent depth maps that are scale-free, so we align this prediction to the current UniDepthV2 metric prediction using (2) by jointly fitting a scale factor α^* and affine correction b^*.

$$(\alpha^*, b^*) = \arg\min_{\alpha,b} \|\mathbf{d}_{\text{uni}} - (\alpha \cdot \mathbf{d}_{\text{vggt}} + b)\|_2^2 \tag{2}$$

However, a single global scale factor is insufficient because the two models' depth errors differ across spatial frequency. We empirically observe that UniDepthV2 has lower error at coarse spatial scales, while VGGT has lower error at fine spatial scales due to multi-view triangulation. Let us define *Spatial Depth Frequency* as the spatial rate of variation of a predicted depth field over image coordinates, and let $B_k(\mathbf{d})$ denote the band-pass component of depth field $\mathbf{d}$ at scale k, obtained by differencing successive Gaussian blurs of $\mathbf{d}$ over N_b increasing scales (we use $N_b = 7$), with B_{N_b+1} the residual low-pass band. Our method computes cross-model reliability at each keyframe using a spatial depth frequency-dependent weight vector $\mathbf{w} = (w_1, \ldots, w_{N_b+1})$. Given the aligned VGGT depth $\hat{\mathbf{d}}_{\text{vggt}}$ from Eq. (2), the fused depth target is

$$\mathbf{d}_{\text{fused}}(\mathbf{w}) = \sum_{k=1}^{N_b+1} \Big[B_k(\mathbf{d}_{\text{uni}}) + w_k \big(B_k(\hat{\mathbf{d}}_{\text{vggt}}) - B_k(\mathbf{d}_{\text{uni}}) \big) \Big]. \tag{3}$$

$w_k = 0$ selects UniDepthV2 for band k while $w_k = 1$ selects aligned VGGT, with intermediate values interpolating between the two. We estimate $\mathbf{w}$ by using the tracker's frozen pose estimate to reproject $\mathbf{d}_{\text{fused}}(\mathbf{w})$ into the previous keyframe's view and compare it against the previous committed fused depth. Before measuring this residual, we remove the residual global scale and offset mismatch via an auxiliary affine fit, so that a uniform scale error is not mistaken for depth-dependent shape error. We regularize $\mathbf{w}$ towards the previous keyframe's weight vector $\mathbf{w}_{t-1}$ so that band weights vary smoothly across keyframes, while still constraining them to $[0, 1]$:

$$\mathbf{w}_t = \arg\min_{\mathbf{w} \in [0,1]^{N_b+1}} L_{\text{geo}}(\mathbf{w}) + \lambda \|\mathbf{w} - \mathbf{w}_{t-1}\|_2^2 . \tag{4}$$

In our implementation, we solve this optimization via zeroth-order coordinate search over L_{geo}, without a photometric term.

4.4 Semantic Metric-Level Correction

Depth fusion addresses differences in spatial structure between UniDepthV2 and VGGT, but it does not address frame-dependent scale variation in UniDepthV2's single-frame predictions. We therefore use objects already accumulated in the map as persistent references for estimating temporal deviations in metric scale.

Obtaining this signal requires persistent object correspondences rather than raw semantic labels. At each keyframe kf_t, we apply Grounding DINO (Liu et al., 2024), followed by box-prompted SAM2 (Ravi et al., 2024), to obtain scored instance masks. We associate instances across keyframes using VGGT point-track correspondences and a lightweight image-space Kalman tracker. A chi-square innovation gate rejects ambiguous associations between objects with the same semantic label.

Each Gaussian inserted between keyframes kf_{t-1} and kf_t inherits the identity of the track-validated instance containing its birth pixel. A semantic landmark is therefore represented by a subset of the map's Gaussians rather than a separate geometric state. Gaussian additions are grouped by keyframe, and inconsistent groups are rejected before the remaining observations are combined. For each visible mature landmark, we compute the median ratio between its rendered depth and the current UniDepthV2 prediction. This ratio measures frame-dependent scale variation relative to the accumulated map, and semantics provide the correspondence needed to construct the measurement, rather than an independent source of metric depth.

The accumulated map also contains uncertainty from previous depth and pose estimates, so landmark ratios cannot be applied directly. We assign each category a measurement variance based on the number and agreement of its landmark observations:

$$\sigma^2_{\text{meas},i} = \frac{\kappa^2 \,\mathrm{MMAD}_i^2}{n_i} + \sigma^2_{\text{floor}}, \tag{5}$$

where MMAD_i denotes the mean median absolute deviation for category i, κ scales this dispersion, and σ^2_{floor} is a noise floor. This gives inconsistent categories less weight. To avoid pooling category-dependent errors, we track each category's log-ratio with a state-space Gaussian process using $\sigma^2_{\text{meas},i}$ as its measurement noise. Its state $\mathbf{x}_t$, containing Ornstein–Uhlenbeck and Matérn-3/2 components, is estimated at each keyframe by

$$\mathbf{x}_t = \arg\min_{\mathbf{x}} (\mathbf{x} - A\mathbf{x}_{t-1})^\top \Sigma_t^{-1} (\mathbf{x} - A\mathbf{x}_{t-1}) + \frac{(y_t - H\mathbf{x})^2}{R_t}, \tag{6}$$

where A and H are derived from the underlying Matérn kernels and $\Sigma_t = AP_{t-1}A^\top + \mathbf{Q}_{\mathrm{GP}}$. The resulting category estimates and their uncertainties are fused with a scalar Kalman filter over δ (the log-scale correction):

$$\hat{\delta}_t = \arg\min_{\delta} \frac{(\delta - \hat{\delta}_{t-1})^2}{P_{t-1} + q_\delta} + \frac{(z_t - \delta)^2}{R_t}, \tag{7}$$

where z_t is the fused measurement across visible categories and R_t is its variance. The semantic correction $\eta_t = \exp(\hat{\delta}_t)$ is estimated at keyframes, then applied to the UniDepthV2 depth predictions of subsequent frames until the next keyframe update, with a gradual decay toward 1 between updates. Appendix A.4 provides the complete filter updates.

4.5 Semantic Loop Closure

At each keyframe after semantic landmarks have been updated, the persistent object-level representation gives us a pose-independent mechanism for identifying loop closures. Each semantic landmark has a corrected depth, so a revisited location stays consistent even after the estimated pose has drifted. Once semantic agreement, obtained through label consistency and pairwise correspondence, identifies candidates, we use VGGT to estimate a relative pose between the current keyframe and the previously observed keyframe. We then verify each candidate geometrically, accepting the candidate only if VGGT triangulates the pair with sufficient confidence and its relative pose prediction disagrees with the current pose estimate by more than a minimum drift threshold.

When a loop closure is verified, we apply its pose correction to the previously constructed Gaussian submap for that region, rather than to only the current frame, similar to (Zhu et al., 2025). Each submap's anchor frame is $a_j = j \cdot \texttt{submap_size}$, where j is the submap index and `submap_size` is a user-set parameter. The correction is a single rigid transform between that anchor's extrinsic matrix before and after the factor-graph update,

$$T_j^{\text{loop}} = E_{a_j}^{\text{new}} \left(E_{a_j}^{\text{old}} \right)^{-1}, \tag{8}$$

which is applied to every Gaussian in the submap. Subsequent tracking and mapping then use this corrected submap geometry. We show the overall methodology of how our algorithm uses semantics in Fig. 4.

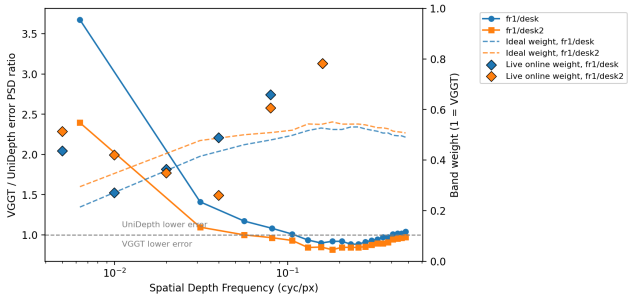

Figure 5: **Depth error spectral analysis between foundation models.** Ratio of VGGT's to UniDepthV2's error power spectral density vs. spatial depth frequency (cyc/px = cycles per pixel, a spatial-frequency measure of image sharpness), with estimated per-band weight w_k overlaid (right axis, 1 = VGGT). Assuming independent model errors within each frequency band, the minimum-variance VGGT weight is $w_k^* = 1/(1 + r_k)$.

4.6 Pose Estimation with VGGT

VGGT provides pose information only at keyframes, while Adam tracking estimates camera motion for every incoming frame. At each keyframe, we initialize the Adam pose optimization from this VGGT prediction rather than extrapolating the previous two poses. We only use it when it agrees with the extrapolated estimate within a fixed translation and rotation bound. Otherwise we fall back to the extrapolated estimate, since VGGT occasionally produces implausible predictions. Once tracking converges, we also add VGGT's measured relative pose between the two most recent keyframes in the window as an extra edge in the factor graph (Sec. A.3), alongside the edge connecting every consecutive pair of frames from Adam's tracked motion. The pose graph then solves for the pose estimate using iSAM2.

5 Experimental Results

5.1 Experimental Setup

We evaluate on TUM RGB-D (Sturm et al., 2012), ScanNet (Dai et al., 2017), and 7-Scenes (Glocker et al., 2013) using one NVIDIA RTX A6000 GPU and the MonoGS post-SLAM refinement conventions (Matsuki et al., 2024). TUM RGB-D provides motion-capture poses, ScanNet uses BundleFusion poses, and 7-Scenes contains handheld Kinect sequences with camera parameters following (Li et al., 2026). We report PSNR, SSIM, and LPIPS in Appendix A.5.4. Sim(3) Absolute Trajectory Error (ATE) compares against monocular baselines after global scale alignment, whereas SE(3) ATE preserves metric scale for comparison with RGB-D systems. Our metric-level poses support SE(3) evaluation without fitting another global scale. Appendix A.5.1 gives the precise formulas.

5.2 Depth Model Fusion Validation

Figure 5 verifies that UniDepthV2 and VGGT possess complementary error structure across spatial depth-frequency bands, motivating online fusion. We measure the error power spectrum of each model against ground truth, after affine-aligning VGGT depth, and plot the resulting ratio against spatial frequency on a logarithmic axis. UniDepthV2 has lower error at lower frequencies, while VGGT is stronger at higher frequencies. The estimated per-band weights w_k follow this trend, crossing 0.5 near the region where the error ratio crosses 1. Importantly, these weights are estimated online through Eq. 4 without access to ground-truth error spectra.

5.3 Semantic Scale Correction Validation

Table 1 reports the Pearson correlation (see Appendix A.5.1), which validates that the landmark ratio contains information about temporal scale variation. For each keyframe, we compute a ground-truth scale factor and remove slow-varying bias and drift to isolate frame-to-frame fluctuations. We then

correlate this short-timescale variation with the raw landmark ratio, which tracks the frame-to-frame fluctuations at $r = 0.56$–0.68 on TUM and $r = 0.28$ on the more cluttered ScanNet scene; the correlation strengthens with more identified landmarks.

Table 1: **Landmark ratio tracks its intended target.** Pearson correlation (log space) between the per-keyframe landmark ratio and a ground-truth scale factor.

Sequence	r, raw	r, frame-to-frame component ($n_{\text{lm}} \geq 4$)
TUM *fr1/desk*	0.43	0.68
TUM *fr1/desk2*	0.42	0.56
ScanNet scene *0059*	0.14	0.28

5.4 Pose and Depth Estimation Performance

Table 2: **TUM RGB-D ATE RMSE [cm].** SE(3) ATE compares to RGB-D baselines; Sim(3) compares to monocular baselines. Baseline values are from Splat-SLAM (Sandström et al., 2025) and Loop-Splat (Zhu et al., 2025). Our fitted scale s for Sim(3) is reported below. Best 1st, 2nd, 3rd.

Method	*fr1/desk*	*fr2/xyz*	*fr3/office*	*fr1/desk2*	*fr1/room*
SE(3), vs. RGB-D					
SplaTAM	3.4	1.2	5.2	6.5	11.1
GS-SLAM	1.5	1.6	1.7	–	–
GO-SLAM (RGB-D)	1.5	0.6	1.3	–	4.7
MonoGS (RGB-D)	1.4	1.4	1.5	5.1	6.3
LoopSplat	2.08	1.58	3.22	3.54	6.24
Ours (SE(3))	4.51	2.93	14.32	4.87	44.52
Sim(3), vs. monocular					
Photo-SLAM	1.5	1.0	1.3	–	–
MonoGS	3.8	5.2	2.9	75.7	76.6
Splat-SLAM	1.6	0.2	1.4	2.8	4.2
Ours (Sim(3))	4.47	2.93	6.64	4.06	41.04
Ours: scale s	0.9926	1.0046	0.9535	1.0267	1.1027

Table 3: **ScanNet ATE RMSE [cm].** SE(3) ATE compares to RGB-D baselines; Sim(3) ATE compares to monocular baselines. Baseline values are from Splat-SLAM (Sandström et al., 2025) and Loop-Splat (Zhu et al., 2025). Our fitted scale s for Sim(3) is reported below. Ours failed on scene *0181*. Best 1st, 2nd, 3rd.

Method	*0000*	*0059*	*0106*	*0169*	*0181*	*0207*
SE(3), vs. RGB-D						
SplaTAM	12.8	10.1	17.7	12.1	11.1	7.5
GS-SLAM	21.2	12.8	13.5	16.3	21.0	14.3
GO-SLAM (RGB-D)	–	–	–	–	–	–
MonoGS (RGB-D)	16.1	6.4	8.1	8.7	26.4	9.2
LoopSplat	6.2	7.1	7.4	10.6	8.5	6.6
Ours (SE(3))	171.87	16.34	16.93	30.17	x	222.46
Sim(3), vs. monocular						
Photo-SLAM	–	–	–	–	–	–
MonoGS	149.2	96.8	155.5	140.3	92.6	101.9
Splat-SLAM	5.5	9.1	7.0	8.2	8.3	7.5
Ours (Sim(3))	160.07	16.23	15.76	29.29	x	222.22
Ours: scale s	1.2308	1.0147	1.022	0.953	x	1.0599

Table 4: **Depth L1 error on TUM RGB-D.**

Scene	*fr1/desk*	*fr2/xyz*	*fr3/office*	*fr1/desk2*	*fr1/room*	Avg.
Depth L1 [cm]	8.29	19.67	28.92	12.84	13.14	**16.57**

Tables 2 and 3 evaluate our trajectory shape against monocular baselines after allowing global scale alignment, while SE(3) ATE retains metric scale. The fitted scale s is reported in the bottom row of each table. On TUM *fr1/desk*, *fr2/xyz*, and *fr1/desk2*, we obtain $s = 0.9926, 1.0046, 1.02667$, showing that the estimated trajectories are already close to metric scale. In SE(3) ATE, we report the second-lowest ATE on *fr1/desk2*. In ScanNet, tracking degrades substantially on scene *0000*, scene *0169*, and scene *0207*, and fails on scene *0181*. However, the fitted scale remains relatively close to one on scene *0169* and scene *0207*, indicating that these failures primarily arise from trajectory drift rather than global scale error. Table 4 separately reports absolute metric reconstruction error directly. Full 7-Scenes results are reported in Appendix A.5.5.

5.5 Loop Closure Analysis

Figure 6 reports the effects of loop closure corrections on the longer ScanNet scene *0059*. Each vertical line marks a keyframe where loop closure fired. Around frame 1590, a cluster of corrections pulls the cumulative Sim(3) ATE RMSE down from a peak of roughly 21 cm; the final full-trajectory

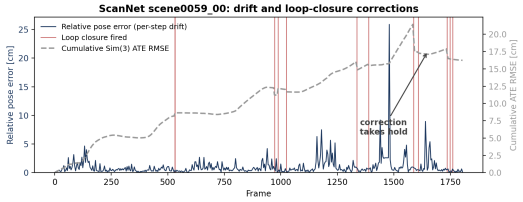

Figure 6: **Drift and loop-closure corrections on ScanNet scene *0059*.** Per-step relative pose error (left axis) and cumulative Sim(3) ATE RMSE (right axis, dashed) against frame index. Red lines mark frames where loop closure fired.

Sim(3) ATE is 16.23 cm, as reported in Table 3. We report loop closure precision and recall in Appendix A.5.9.

5.6 Ablations

Table 5: **Ablation study.** TUM RGB-D *fr1/desk2* sequence. Best 1st , 2nd , 3rd .

Configuration	Sim(3) ATE ↓	PSNR ↑	SSIM ↑	LPIPS ↓	Depth L1 ↓
Full (Ours)	4.06	19.44	0.719	0.396	12.84
No semantics	10.60	17.40	0.689	0.441	15.24
No depth fusion	4.59	19.26	0.714	0.402	13.33
No loop closure	7.39	17.37	0.692	0.426	15.19

Table 5 reports ablations, isolating the contribution of each major component of our system on TUM *fr1/desk2*. We see that removing semantic processing produced the largest increase in trajectory error (4.06–10.60 cm), while removing loop closures increased Sim(3) ATE to 7.39 cm and reduced rendering performance. Removing the frequency-dependent depth fusion produced smaller but consistent degradation, while VGGT-derived point-track correspondences and pose estimates remained active.

5.7 Limitations

Our current implementation has substantial runtimes, largely due to expensive mapping, and tracking steps (see Appendix A.5.7 for detailed timings). Metric accuracy inherits frame-dependent scale variation from UniDepthV2, although our failure analysis in Appendix A.5.2 suggests that this is not the dominant source of the observed ScanNet tracking failures. VGGT can also become unreliable in scenes without sufficient parallax (see Appendix A.5.3), which can affect both pose estimation and geometric consistency in the map. Furthermore, loop closure requires repeat observations with both semantically and geometrically verifiable matches, so our approach weakens on scenes or position revisits with few objects.

6 Conclusion

We introduce S^3-SLAM, a training-free monocular 3DGS SLAM system capable of estimating multi-view-consistent metric-level depth. Our results show that UniDepthV2 and VGGT possess complementary error structure across spatial depth frequencies, while semantic landmarks provide a mechanism to correct short-timescale metric-scale variation. Future work will focus on improving pose-estimation robustness and computational efficiency, extending the framework to dynamic SLAM, and applying similar methods to egocentric video.

AI use statement

Generative AI tools were used to help generate mathematical formulations, iterate on algorithm design, provide feedback on hyperparameter tuning, and data analysis. We also used these tools to

help with initial idea and paper section drafting. We have reviewed all AI-assisted work and take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

Reproducibility Statement

We plan to release our code as open source on GitHub, as well as develop a project page with a summary and instructions on how to run our algorithm.

References

Chong Cheng, Sicheng Yu, Zijian Wang, Yifan Zhou, and Hao Wang. Outdoor monocular slam with global scale-consistent 3d gaussian pointmaps. *arXiv preprint arXiv:2507.03737*, 2025.

Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017.

Andrew J. Davison, Ian D. Reid, Nicholas D. Molton, and Olivier Stasse. Monoslam: Real-time single camera slam. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6): 1052–1067, 2007. doi: 10.1109/TPAMI.2007.1049.

Frank Dellaert. Factor Graphs: Exploiting Structure in Robotics. *Annual Review of Control, Robotics, and Autonomous Systems*, 4(1):141–166, 2021.

Frank Dellaert and Michael Kaess. *Factor Graphs for Robot Perception*. Now Publishers Inc., Hanover, MA, USA, 2017. ISBN 168083326X.

Abdelrahman Eldesokey, Michael Felsberg, and Fahad Shahbaz Khan. Confidence propagation through cnns for guided sparse depth regression. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(10):2423–2436, October 2020. ISSN 1939-3539. doi: 10.1109/tpami.2019.2929170. URL http://dx.doi.org/10.1109/TPAMI.2019.2929170.

Lu Gan, Maani Ghaffari Jadidi, Steven A. Parkison, and Ryan M. Eustice. Sparse bayesian inference for dense semantic mapping, 2017. URL https://arxiv.org/abs/1709.07973.

Lu Gan, Youngji Kim, Jessy W. Grizzle, Jeffrey M. Walls, Ayoung Kim, Ryan M. Eustice, and Maani Ghaffari. Multitask learning for scalable and dense multilayer bayesian map inference. *IEEE Transactions on Robotics*, 39(1):699–717, 2023. doi: 10.1109/TRO.2022.3197106.

Ben Glocker, Shahram Izadi, Jamie Shotton, and Antonio Criminisi. Real-time RGB-D camera relocalization. In *Proceedings of the IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, 2013.

Wolfgang Hess, Damon Kohler, Holger Rapp, and Daniel Andor. Real-time loop closure in 2d lidar slam. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1271–1278, 2016. doi: 10.1109/ICRA.2016.7487258.

Huajian Huang, Longwei Li, Hui Cheng, and Sai-Kit Yeung. Photo-slam: Real-time simultaneous localization and photorealistic mapping for monocular, stereo, and rgb-d cameras. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21584–21593, 2024. doi: 10.1109/CVPR52733.2024.02039.

Michael Kaess, Hordur Johannsson, Richard Roberts, Viorela Ila, John Leonard, and Frank Dellaert. isam2: Incremental smoothing and mapping with fluid relinearization and incremental variable reordering. In *2011 IEEE International Conference on Robotics and Automation*, pp. 3281–3288, 2011. doi: 10.1109/ICRA.2011.5979641.

Michael Kaess, Hordur Johannsson, Richard Roberts, Viorela Ila, John J Leonard, and Frank Dellaert. isam2: Incremental smoothing and mapping using the bayes tree. *Int. J. Rob. Res.*, 31(2):216–235, February 2012. ISSN 0278-3649. doi: 10.1177/0278364911430419. URL https://doi.org/10.1177/0278364911430419.

Nikhil Keetha, Jay Karhade, Krishna Murthy Jatavallabhula, Gengshan Yang, Sebastian Scherer, Deva Ramanan, and Jonathon Luiten. *SplaTAM: Splat, Track & Map 3D Gaussians for Dense RGB-D SLAM*. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. URL https://arxiv.org/abs/2312.02126. ArXiv preprint arXiv:2312.02126.

B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graphics*, 42(4), July 2023. doi: 10.1145/3592433. URL https://dl.acm.org/doi/10.1145/3592433.

Junhyung Kim and Jinwhan Kim. Semantic loop closure for reducing false matches in slam. In *2025 22nd International Conference on Ubiquitous Robots (UR)*, pp. 34–39, 2025. doi: 10.1109/UR65550.2025.11077984.

J.J. Leonard and H.F. Durrant-Whyte. Mobile robot localization by tracking geometric beacons. *IEEE Transactions on Robotics and Automation*, 7(3):376–382, 1991. doi: 10.1109/70.88147.

Kunyi Li, Michael Niemeyer, Sen Wang, Stefano Gasperini, Nassir Navab, and Federico Tombari. Sing3r-slam: Submap-based indoor monocular gaussian slam with 3d reconstruction priors. *IEEE Robotics and Automation Letters*, pp. 1–8, 2026. doi: 10.1109/LRA.2026.3734819.

Seunghyun Lim, Jeyoung Park, Junmo Yang, and Dongwon Yun. mono-morp: Monocular traversability representation and motion planner for topological visual navigation. *IEEE Robotics and Automation Letters*, 11(10):12144–12151, 2026. doi: 10.1109/LRA.2026.3728308.

Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, and Lei Zhang. Grounding DINO: Marrying DINO with grounded pre-training for open-set object detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 38–55, 2024. ArXiv preprint arXiv:2303.05499.

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

Dominic Maggio, Hyungtae Lim, and Luca Carlone. Vggt-slam: Dense rgb slam optimized on the sl (4) manifold. *Advances in Neural Information Processing Systems*, 39, 2025.

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

Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: representing scenes as neural radiance fields for view synthesis. *Commun. ACM*, 65(1):99–106, December 2021. ISSN 0001-0782. doi: 10.1145/3503250. URL https://doi.org/10.1145/3503250.

Mustafa Mukadam, Jing Dong, Frank Dellaert, and Byron Boots. STEAP: simultaneous trajectory estimation and planning. *CoRR*, abs/1807.10425, 2018. URL http://arxiv.org/abs/1807.10425.

Raúl Mur-Artal, J. M. M. Montiel, and Juan D. Tardós. Orb-slam: A versatile and accurate monocular slam system. *IEEE Transactions on Robotics*, 31(5):1147–1163, 2015. doi: 10.1109/TRO.2015.2463671.

Luigi Piccinelli, Christos Sakaridis, Yung-Hsu Yang, Mattia Segu, Siyuan Li, Wim Abbeloos, and Luc Van Gool. Unidepthv2: Universal monocular metric depth estimation made simpler. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 48(3):2354–2367, 2026. doi: 10.1109/TPAMI.2025.3628473.

Huayan Pu, Jun Luo, Gang Wang, Tao Huang, Hongliang Liu, and Jun Luo. Visual slam integration with semantic segmentation and deep learning: A review. *IEEE Sensors Journal*, 23(19):22119–22138, 2023. doi: 10.1109/JSEN.2023.3306371.

Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. SAM 2: Segment anything in images and videos, 2024. URL https://arxiv.org/abs/2408.00714.

Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *Proceedings of the 2011 International Conference on Computer Vision*, ICCV '11, pp. 2564–2571, USA, 2011. IEEE Computer Society. ISBN 9781457711015. doi: 10.1109/ICCV.2011.6126544. URL https://doi.org/10.1109/ICCV.2011.6126544.

Erika Sandström, Yue Li, Luc Van Gool, and Martin R. Oswald. Point-slam: Dense neural point cloud-based slam. *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 18387–18398, 2023. URL https://api.semanticscholar.org/CorpusID:258049300.

Erik Sandström, Ganlin Zhang, Keisuke Tateno, Michael Oechsle, Michael Niemeyer, Youmin Zhang, Manthan Patel, Luc Van Gool, Martin R. Oswald, and Federico Tombari. Splat-slam: Globally optimized rgb-only slam with 3d gaussians. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 1671–1682, 2025. doi: 10.1109/CVPRW67362.2025.00156.

J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers. A benchmark for the evaluation of rgb-d slam systems. In *Proc. of the International Conference on Intelligent Robot Systems (IROS)*, Oct. 2012.

Fabio Tosi, Youmin Zhang, Ziren Gong, Erik Sandström, Stefano Mattoccia, Martin R. Oswald, and Matteo Poggi. How nerfs and 3-d gaussian splatting are reshaping slam: A survey. *IEEE Transactions on Robotics*, 42:1405–1427, 2026. doi: 10.1109/TRO.2026.3666139.

Dario Tscholl, Yashwanth Kumar Nakka, and Brian Gunter. Fastbridge: Closing the model-based realization gap in safety filters on 3d gaussian splatting for fast quadrotor flight, 2026. URL https://arxiv.org/abs/2607.01200.

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

Stephan Weiss, Davide Scaramuzza, and Roland Siegwart. Monocular-slam–based navigation for autonomous micro helicopters in gps-denied environments. *Journal of Field Robotics*, 28, 2011. URL https://api.semanticscholar.org/CorpusID:2085968.

Chi Yan, Delin Qu, Dan Xu, Bin Zhao, Zhigang Wang, Dong Wang, and Xuelong Li. Gs-slam: Dense visual slam with 3d gaussian splatting, 2024. URL https://arxiv.org/abs/2311.11700.

Sicheng Yu, Chong Cheng, Yifan Zhou, Xiaojun Yang, and Hao Wang. Rgb-only gaussian splatting slam for unbounded outdoor scenes. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 11068–11074, 2025. doi: 10.1109/ICRA55743.2025.11128286.

Ganlin Zhang, Erik Sandström, Youmin Zhang, Manthan Patel, Luc Van Gool, and Martin R. Oswald. Glorie-slam: Globally optimized rgb-only implicit encoding point cloud slam, 2024. URL https://arxiv.org/abs/2403.19549.

Liang Zhang, Leqi Wei, Peiyi Shen, Wei Wei, Guangming Zhu, and Juan Song. Semantic slam based on object detection and improved octomap. *IEEE Access*, 6:75545–75559, 2018. doi: 10.1109/ACCESS.2018.2873617.

Wei Zhang, Qing Cheng, David Skuddis, Niclas Zeller, Daniel Cremers, and Norbert Haala. Hi-slam2: Geometry-aware gaussian slam for fast monocular scene reconstruction. *IEEE Transactions on Robotics*, 41:6478–6493, 2025. ISSN 1941-0468. doi: 10.1109/tro.2025.3626627. URL http://dx.doi.org/10.1109/TRO.2025.3626627.

Youmin Zhang, Fabio Tosi, Stefano Mattoccia, and Matteo Poggi. Go-slam: Global optimization for consistent 3d instant reconstruction, 2023. URL https://arxiv.org/abs/2309.02436.

Liyuan Zhu, Yue Li, Erik Sandström, Shengyu Huang, Konrad Schindler, and Iro Armeni. Loopsplat: Loop closure by registering 3d gaussian splats. In *International Conference on 3D Vision (3DV)*, 2025.

A Appendix

A.1 SplaTAM Conventions

A.1.1 3DGS Implementation

SplaTAM (Keetha et al., 2024) is an RGB-D 3DGS SLAM algorithm that uses differentiable rendering via splatting for both color and depth, as shown in Eqs. (9)–(11), which define the rendered color (C), depth (D), and silhouette (S), respectively, of an input pixel $\mathbf{p}$,

$$C(\mathbf{p}) = \sum_{i=1}^{n} c_i f_i(\mathbf{p}) \prod_{j=1}^{i-1} (1 - f_j(\mathbf{p})) \tag{9}$$

$$D(\mathbf{p}) = \sum_{i=1}^{n} d_i f_i(\mathbf{p}) \prod_{j=1}^{i-1} (1 - f_j(\mathbf{p})), \tag{10}$$

$$S(\mathbf{p}) = \sum_{i=1}^{n} f_i(\mathbf{p}) \prod_{j=1}^{i-1} (1 - f_j(\mathbf{p})), \tag{11}$$

where c_i and d_i are the color and camera-space depth of Gaussian i, respectively, and $f_i(\mathbf{p})$ is its opacity-weighted contribution at pixel $\mathbf{p}$. Under this isotropic formulation, $f_i(\mathbf{p})$ is computed by the following unnormalized Gaussian equation weighted by opacity,

$$f_i(\mathbf{p}) = o_i \exp\left(-\frac{\left\|\mathbf{p} - \boldsymbol{\mu}_i^{2D}\right\|^2}{2\left(r_i^{2D}\right)^2} \right), \tag{12}$$

with $o_i \in [0, 1]$ being opacity, $\boldsymbol{\mu}_i^{2D}$ being the projected Gaussian center, and r_i^{2D} being the projected scalar isotropic radius. The 3D center and radius are projected into pixel space by

$$\boldsymbol{\mu}_i^{2D} = K \frac{E_t \boldsymbol{\mu}_i^{3D}}{d_i}, \qquad r_i^{2D} = \frac{f r_i}{d_i}, \quad \text{where } d_i = \left(E_t \boldsymbol{\mu}_i^{3D}\right)_z, \tag{13}$$

where K is the camera calibration matrix, E_t is the camera extrinsic matrix, f is the focal length, and d_i is the depth of Gaussian i in camera coordinates. We use this RGB-D architecture as our base because it directly incorporates depth supervision into the mapping process.

A.1.2 Depth Map Input and SLAM Process

Each frame is processed through three stages: tracking, mapping, and densification, operating on the set of Gaussians defined in A.1.1.

The *Tracking* process estimates the current camera pose by minimizing photometric and depth rerendering loss via Adam optimization. In our approach, the process of how this is integrated with VGGT pose estimates in the factor graph at keyframes is discussed in 4.6.

Mapping then optimizes the existing Gaussian parameters by rendering from the current pose and minimizing depth and photometric loss, while periodically pruning low-opacity Gaussians.

Densification inserts new Gaussians by back-projecting the current depth map if either the rendered silhouette (which holds the accumulated opacity coverage) falls below a set threshold, or if the rendered depth significantly differs from the depth target (which is obtained from raw UniDepthV2 in normal frames, or the fused VGGT-UniDepthV2 on keyframes). These conditions indicate that an occluding surface is not properly captured.

A.2 Visual Geometry Grounded Transformers (VGGT)

VGGT maps an image sequence to per-image predictions:

$$f\big(\{I_i\}_{i=1}^N\big) = \{(g_i, D_i, P_i, T_i)\}_{i=1}^N, \tag{14}$$

where $g_i \in \mathbb{R}^9$ contains the camera parameters, $D_i \in \mathbb{R}^{H\times W}$ is the depth map, $P_i \in \mathbb{R}^{3\times H\times W}$ is the point map, and $T_i \in \mathbb{R}^{C\times H\times W}$ is a grid of features used for point tracking.

A.3 Factor Graphs

A factor graph is a bipartite graph $F = (\mathcal{U}, \mathcal{V}, \mathcal{E})$ (Dellaert & Kaess, 2017), where $\phi_i \in \mathcal{U}$ represents a factor node, $x_j \in \mathcal{V}$ represents a variable node, and $e_{ij} \in \mathcal{E}$ represents a dependency of factor ϕ_i on variable x_j. This defines the factorization of a global function $\phi(X)$ as

$$\phi(X) = \prod_i \phi_i(X_i), \tag{15}$$

where $X_i \subseteq X$ denotes the subset of variables relevant to factor ϕ_i.

The iSAM2 algorithm (Kaess et al., 2012) enables this optimization problem to be updated incrementally as new measurements and variables are added. Internally, iSAM2 maintains a Bayes tree and recomputes only the affected cliques when new information is added. Factor-graph methods have been broadly applied to VIO and SLAM (Dellaert, 2021), as well as online motion planning (Mukadam et al., 2018).

A.4 Semantic Tracking

This section gives the full update equations for the two filters used in Sec. 4.4 to estimate the semantic metric-level correction: a per-category state-space Gaussian process (SSGP), and a scalar Kalman filter that fuses its outputs across categories.

A.4.1 Per-Category Level Estimation with an SSGP

For each semantic category i, we track zero-mean Matérn-1/2 (Ornstein–Uhlenbeck) and Matérn-3/2 components. The state, measurement model, and prediction over a step dt are

$$\mathbf{x}_t = [\mathrm{OU}_t, \mathrm{M32}_t, \mathrm{M32}'_t]^\top. \tag{16}$$

$$H = [1, 1, 0]. \tag{17}$$

$$y_t = H\mathbf{x}_t + \epsilon_t. \tag{18}$$

$$R_t = \sigma^2_{\text{meas},i}. \tag{19}$$

$$\mathbf{x}_t^- = A\mathbf{x}_{t-1}. \tag{20}$$

$$P_t^- = AP_{t-1}A^\top + \mathbf{Q}_{\mathrm{GP}}. \tag{21}$$

where R_t is given by Eq. (5). The matrices A and $\mathbf{Q}_{\mathrm{GP}}$ are block-diagonal, with exact Ornstein–Uhlenbeck and Matérn-3/2 discretizations at step dt. When a measurement y_t is available, the filter updates

$$S_t = HP_t^-H^\top + R_t. \tag{22}$$

$$G_t = \frac{P_t^-H^\top}{S_t}. \tag{23}$$

$$\mathbf{x}_t = \mathbf{x}_t^- + G_t(y_t - H\mathbf{x}_t^-). \tag{24}$$

$$P_t = (I - G_tH)P_t^-(I - G_tH)^\top + G_tG_t^\top R_t; \tag{25}$$

otherwise $\mathbf{x}_t = \mathbf{x}_t^-$, $P_t = P_t^-$. The per-category output consumed by the fusion filter below is

$$\eta_{\mathrm{OU},i} = \exp\left([\mathbf{x}_t]_1\right). \tag{26}$$

$$\sigma_{\mathrm{OU},i} = \sqrt{[P_t]_{11}}. \tag{27}$$

A.4.2 Cross-Category Scale Fusion (Innovation-Gated Kalman Filter)

At each keyframe kf_t, categories with sufficient landmark evidence are fused using log-ratio measurements and inverse-variance weights:

$$\ell_i = \log \eta_{\mathrm{OU},i}. \tag{28}$$

$$u_i = \frac{n_{\mathrm{used},i}}{\sigma^2_{\mathrm{OU},i}}. \tag{29}$$

$$z_t = \frac{\sum_i \ell_i u_i}{\sum_i u_i}. \tag{30}$$

$$R_t = \left(\sum_i u_i \right)^{-1}. \tag{31}$$

The filter predicts the prior variance under a constant-level model with process variance q_δ, then forms the innovation and its variance:

$$P_t^- = P_{t-1} + q_\delta. \tag{32}$$

$$\nu_t = z_t - \hat{\delta}_{t-1}. \tag{33}$$

$$S_t = P_t^- + R_t. \tag{34}$$

The measurement is accepted only if it is statistically consistent with the current state estimate under a chi-square test with one degree of freedom,

$$\frac{\nu_t^2}{S_t} < \chi^2_{\mathrm{thr}}. \tag{35}$$

If accepted, the state and variance are updated via the standard Kalman gain,

$$G_t = \frac{P_t^-}{S_t}. \tag{36}$$

$$\hat{\delta}_t = \hat{\delta}_{t-1} + G_t \nu_t. \tag{37}$$

$$P_t = (1 - G_t) P_t^-; \tag{38}$$

otherwise the measurement is rejected and the state holds, $\hat{\delta}_t = \hat{\delta}_{t-1}$, $P_t = P_t^-$. The semantic scale correction applied to the depth target is

$$\eta_t = \mathrm{clip}\left(\exp(\hat{\delta}_t),\ \frac{1}{1 + \Delta_{\max}},\ 1 + \Delta_{\max} \right). \tag{39}$$

A.5 Results

This section gives the metric formulas used in Sec. 5.1, along with rendering results, the full 7-Scenes results, detailed timing analysis, depth evaluation against Splat-SLAM's protocol, and the loop closure precision/recall table.

A.5.1 Evaluation Metrics

For a trajectory of N poses, let $\mathbf{p}_t$ and $\mathbf{p}_t^{gt}$ denote the estimated and ground-truth camera positions at frame t (Sturm et al., 2012). ATE RMSE aligns the estimated trajectory to ground truth and measures the residual translation error,

$$\text{ATE RMSE} = \sqrt{\frac{1}{N} \sum_{t=1}^{N} \left\| \mathbf{p}_t^{gt} - (sR\mathbf{p}_t + \mathbf{t}) \right\|^2}, \tag{40}$$

where $(s, R, \mathbf{t})$ are the scale, rotation, and translation that minimize this residual. Sim(3) ATE solves for s jointly with R and $\mathbf{t}$. SE(3) ATE fixes $s = 1$.

Following the convention in Point-SLAM (Sandström et al., 2023), the PSNR between a rendered image I and its ground truth I^{gt} is

$$\text{PSNR}(I, I^{gt}) = 10 \log_{10} \left(\frac{1}{\text{MSE}(I, I^{gt})} \right), \qquad \text{MSE}(I, I^{gt}) = \frac{1}{HW} \sum_{p} \left\| I_p - I_p^{gt} \right\|^2, \tag{41}$$

with pixel values normalized to $[0, 1]$. SSIM is computed over local windows x, y of I and I^{gt},

$$\text{SSIM}(x, y) = \frac{(2\mu_x \mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}, \tag{42}$$

and LPIPS is the weighted distance between learned feature activations ϕ_l of a pretrained network at layer l,

$$\text{LPIPS}(I, I^{gt}) = \sum_{l} \frac{1}{H_l W_l} \sum_{p} \left\| \omega_l \odot \left(\phi_l(I)_p - \phi_l(I^{gt})_p \right) \right\|_2^2. \tag{43}$$

Following the convention in (Eldesokey et al., 2020), who correlate a predicted per-pixel confidence signal against ground-truth depth error, we validate the semantic scale correction in Sec. 5.3 with the Pearson correlation coefficient between two scalar sequences X and Y,

$$\rho_{X,Y} = \frac{\text{cov}(X, Y)}{\sigma_X \sigma_Y}. \tag{44}$$

The two depth sources solve different problems and therefore fail in different ways: UniDepthV2 supplies frame-wise metric scale, while VGGT supplies multi-view geometric consistency. We analyze the ScanNet failures in Table 3 (scene *0000*, scene *0181*, and scene *0207*) to determine which model is responsible, and whether our fusion and correction mechanisms can suppress the resulting error.



Figure 7: **Failure examples on ScanNet.** (Top) Ground-truth RGB, rasterized RGB, per-pixel depth error for a typical frame. (Middle) Two identified failure modes: a weak-parallax and textureless wall that causes degeneracy in VGGT's motion estimate. (Bottom) Pose has diverged but the locally rendered depth is still accurate, while the map has become degenerate.

A.5.2 UniDepthV2 Failure Modes

We do not find UniDepthV2's per-frame depth quality to be the dominant driver of the ScanNet failures. In scene *0000*'s divergence window (Fig. 7, bottom), Depth L1 stays low (0.15m) even though the trajectory has already diverged by several meters at that point. As observed in our per-frame depth analysis, UniDepthV2 exhibits affine scale variation and a near-range over-prediction bias, which our per-band depth fusion (Sec. 4.3) and semantic scale correction (Sec. 4.4) are built to suppress. This error would have to be substantially larger for UniDepthV2 to drive the observed failure.

A.5.3 VGGT Failure Modes

VGGT's inter-frame motion estimate degenerates under weak parallax, such as when a large, nearby, textureless surface dominates the field of view and gives little multi-view triangulation signal. Fig. 7 (middle) shows a representative case, with a narrow doorway opening onto a flat wall in scene *0207* at t=499. The motion-derived scale ratio r_{motion}, which should sit near 1, spikes to 12–24$\times$ its normal range at this point, and later up to 142$\times$ in scene *0000*, triggering a degenerate motion detection gate.

Effect on fusion. The per-band depth-fusion weight (Sec. 4.3) largely avoids this failure mode, since it reweights the depth target's spatial-frequency content and does not depend on the motion-derived translation scale. The pose side has no equivalent mechanism, so a degenerate VGGT motion window directly corrupts tracking. Recovery then depends on loop closure: in scene *0000*, semantic loop closure fires 16 times and successfully corrects earlier drift (e.g. 0.79m $\rightarrow$ 0.10m near $t\approx$2200–2400), but stops firing once the pose corrupts past a point where our algorithm struggles to find closures. In scene *0207*, loop closure never fires despite up to 87 initial loop-closure candidates per frame, because every candidate is rejected at the semantic verification gate, leaving a permanently uncorrected drift.

A.5.4 Rendering Performance

Table 6: **Rendering performance on TUM RGB-D.** Keyframes only, following Splat-SLAM's protocol. Baselines as reported. Best 1st, 2nd, 3rd.

Method	Metric	*fr1/desk*	*fr2/xyz*	*fr3/office*	*fr1/desk2*	*fr1/room*
Photo-SLAM	PSNR↑	20.97	21.07	19.59	–	–
	SSIM↑	–	–	–	–	–
	LPIPS↓	0.23	0.17	0.24	–	–
MonoGS	PSNR↑	19.67	16.17	20.63	19.16	18.41
	SSIM↑	0.73	0.72	0.77	0.66	0.64
	LPIPS↓	0.33	0.31	0.34	0.48	0.51
GlORIE-SLAM	PSNR↑	20.26	25.62	21.21	19.09	18.78
	SSIM↑	0.79	0.72	0.72	0.92	0.73
	LPIPS↓	0.31	0.09	0.32	0.38	0.38
Splat-SLAM	PSNR↑	25.61	29.53	26.05	23.98	24.06
	SSIM↑	0.84	0.90	0.84	0.81	0.80
	LPIPS↓	0.18	0.08	0.20	0.23	0.24
Ours	PSNR↑	18.92	19.27	18.97	19.44	19.34
	SSIM↑	0.701	0.665	0.690	0.719	0.698
	LPIPS↓	0.398	0.402	0.470	0.396	0.376

Table 6 illustrates the effect of decoupling mapping from tracking: although our Sim(3) ATE on *fr1/room* is among the weakest results in Table 2, our PSNR and LPIPS rank second on that sequence. Note that GLORIE-SLAM (Zhang et al., 2024) and Splat-SLAM have the same tracking, which is why GLORIE-SLAM wasn't compared earlier.

A.5.5 7-Scenes Pose Estimation

Table 7 gives the full per-scene pose results on 7-Scenes using the evaluation protocol described in Sec. 5.1.

Table 7: **Sim(3) ATE RMSE [cm] on 7-Scenes.** Following the evaluation protocol of (Li et al., 2026). Best 1st , 2nd , 3rd .

Method	*chess*	*fire*	*heads*	*office*	*pumpkin*	*redkitchen*	*stairs*
VGGT-SLAM	3.6	2.8	1.8	10.3	13.3	5.8	9.3
HI-SLAM2	3.8	3.1	2.6	8.5	14.2	4.0	2.4
SING3R-SLAM	3.3	2.7	4.5	7.2	8.7	3.6	4.5
Ours	5.72	3.70	3.90	8.45	16.01	5.2	4.1

A.5.6 ScanNet Pose and Rendering Performance

Table 8 gives rendering results on the six ScanNet scenes reported in (Sandström et al., 2025), using their published baseline numbers. Scene *0181* failed for our method.

Table 8: **Rendering performance on ScanNet.** Monocular baselines only, from (Sandström et al., 2025). †Ours excludes scene *0181* (tracking failed). Best 1st , 2nd , 3rd .

Method	Metric	*0000*	*0059*	*0106*	*0169*	*0181*	*0207*	Avg.
MonoGS	PSNR↑	16.91	19.15	18.57	20.21	19.51	18.37	18.79
	SSIM↑	0.62	0.69	0.74	0.74	0.75	0.70	0.71
	LPIPS↓	0.70	0.51	0.55	0.54	0.63	0.58	0.59
GlORIE-SLAM	PSNR↑	23.42	20.66	20.41	25.23	21.28	23.68	22.45
	SSIM↑	0.87	0.87	0.83	0.84	0.91	0.76	0.85
	LPIPS↓	0.26	0.31	0.31	0.21	0.44	0.29	0.30
Splat-SLAM	PSNR↑	28.68	27.69	27.70	31.14	31.15	30.49	29.48
	SSIM↑	0.83	0.87	0.86	0.87	0.84	0.84	0.85
	LPIPS↓	0.19	0.15	0.18	0.15	0.23	0.19	0.18
Ours	PSNR↑	18.91	18.86	19.06	19.67	x	20.37	19.37†
	SSIM↑	0.678	0.704	0.735	0.731	x	0.747	0.719†
	LPIPS↓	0.611	0.524	0.531	0.520	x	0.502	0.538†

A.5.7 Computational Overhead

Figure 8 breaks per-frame wall-clock time down into mapping (60 iterations/frame) and tracking (200 iterations/frame, including VGGT and UniDepthV2 inference) against the number of Gaussians in the map over the ScanNet scene *0059* run. Tracking dominates at roughly 4–6 s/frame against mapping's roughly 1.5–2 s/frame. Both curves spike when loop closure fires and drift upward slightly as the map grows from 0.4M to 1.8M Gaussians over the first 400 frames.

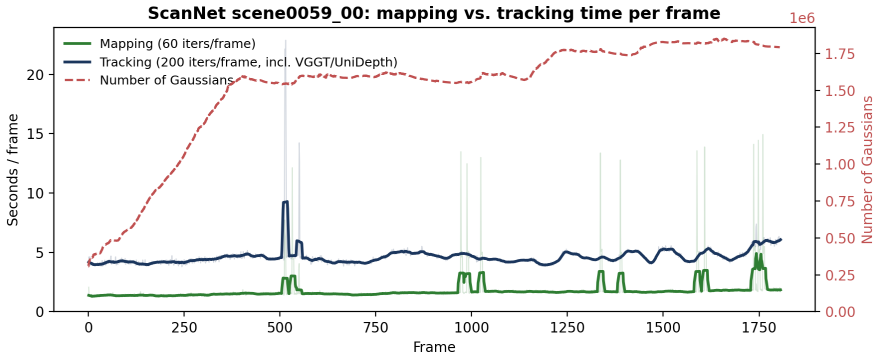

Figure 8: **Mapping vs. tracking time per frame on ScanNet scene *0059*.** Smoothed per-frame wall-clock time for mapping and tracking (left axis) against the number of Gaussians in the map (right axis, dashed) over the course of the run.

Figure 9 isolates components within that tracking total. The tracking total includes VGGT, UniDepthV2, and Adam and is therefore not additive with the component bars. VGGT inference on a 4-frame window costs 0.42s and is essentially constant across scenes (0.40–0.43s mean across five ScanNet scenes), since its cost is set by window size and resolution rather than scene content.

Semantic detection (Grounding DINO + SAM2) costs 1.97s per keyframe. VGGT and semantic detection run only at keyframes, whereas mapping and tracking run on every frame; consequently, the latter dominate total runtime.

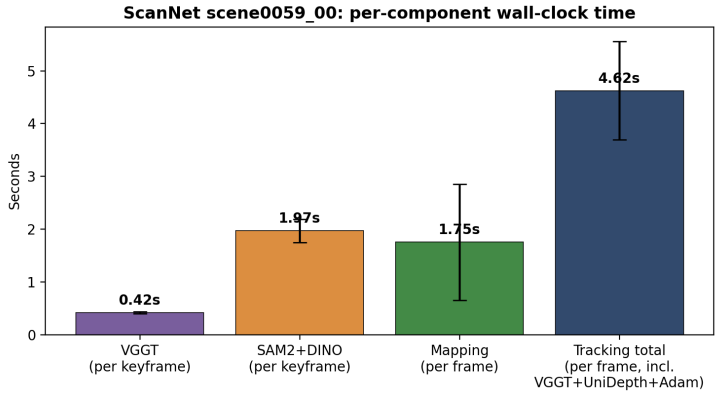

Figure 9: **Per-component wall-clock time on ScanNet scene *0059*.** VGGT and semantic detection (Grounding DINO + SAM2) run once per keyframe, whereas mapping and the tracking total run every frame. The tracking total includes VGGT, UniDepthV2, and Adam, so the bars are not additive. Error bars show one standard deviation.

A.5.8 Depth Evaluation

In Sandström et al. (2025), the authors do not use the ground-truth depth directly when computing Depth L1. Instead, they first compute a value s that scales their Sim(3) trajectory estimate to the ground truth, and then use

$$\text{mean}_t \left[\text{mean}_{p:\, D_{tp}^{GT}>0,\, D_{tp}^{rend}>0} \left| s D_{tp}^{rend} - D_{tp}^{GT} \right| \right]. \tag{45}$$

Table 9 reports the same evaluation after applying Splat-SLAM's Sim(3)-scaled depth protocol. Under this protocol, our average Depth L1 increases from 16.57 cm to 17.07 cm and is slightly higher than Splat-SLAM's 15.78 cm.

Table 9: **Depth L1 error [cm] on TUM RGB-D.** Splat-SLAM's Sim(3)-scaled protocol. Best 1st, 2nd, 3rd.

Method	*fr1/desk*	*fr1/desk2*	*fr1/room*	*fr2/xyz*	*fr3/office*	Avg.
Original	8.29	12.84	13.14	19.67	28.92	16.57
Scale+mask corrected	7.89	11.62	18.26	21.06	26.50	17.07
Splat-SLAM	8.05	15.70	15.05	14.53	25.59	15.78

A.5.9 Loop Closure Precision and Recall

Table 10: **Loop closure precision and recall.** On TUM *fr1/desk* and *fr1/desk2*. Best 1st, 2nd.

Method	Recall ↑	TP density ↑	Precision ↑
Ours (semantics on)	0.500	6.00	0.571
No semantics (pose-proximity only)	0.250	2.00	1.000