PRISM-SLAM: Probabilistic Ray-Grounded Inference for Scale-aware Metric SLAM

Anonymous Authors[1]

Abstract

Monocular SLAM historically suffers from scale ambiguity and tracking failure in dynamic environments. While recent vision foundation models (VFMs) provide remarkable zero-shot depth priors, naively integrating these deterministic predictions ignores predictive uncertainty and frame-to-frame scale inconsistencies. We propose PRISM-SLAM, a real-time framework that rigorously integrates VFM priors into a structured Bayesian factor graph to achieve scale-aware, metric-consistent localization and mapping. Specifically, we introduce a Plücker Ray-Distance Factor to anchor monocular observations in absolute space within a globally consistent metric coordinate system, mathematically resolving scale drift by making the metric scale Fisher-identifiable. To handle environmental dynamics, we derive an epistemic uncertainty proxy from temporal depth consistency and formulate a Dynamic Scene Uncertainty Gating (DSUG) mechanism. This soft-gating approach probabilistically down-weights dynamic distractors without incurring the heavy computational overhead associated with traditional semantic segmentation masks. By employing a multi-process architecture that asynchronously processes VFM inference and geometric tracking, PRISM-SLAM provides verified metric output at 30 FPS using solely RGB input, bridging the gap between foundation models and real-world robotic applications. Evaluated on the TUM RGB-D and 7-Scenes benchmarks, PRISM-SLAM achieves a metric $SE(3)$ Absolute Trajectory Error (ATE) nearly identical to its oracle-aligned $Sim(3)$ error. This demonstrates that our system can produce deployment-ready metric trajectories by delivering robust metric SLAM solutions without any post-hoc scale correction.

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

Project page: https://prismslam-cmd.github.io/prismslam_pr/

1. Introduction

Monocular SLAM is a foundational technology for autonomous driving and robotics due to its minimal hardware constraints. However, traditional geometry-based systems (Campos et al., 2021) are bound by a fundamental mathematical limitation: the unobservability of absolute scale due to the inherent ambiguity of pinhole projective geometry (Hartley & Zisserman, 2003). This critical flaw leads to severe scale drift, a problem further exacerbated in dynamic environments where strict scene rigidity assumptions are routinely violated (Bescos et al., 2018; Zheng et al., 2025).

Recently, Vision Foundation Models (VFMs) have demonstrated remarkable potential in reconstructing 3D structures from single images. While these models offer a path toward metric-consistent SLAM, existing learning-based systems (Teed & Deng, 2021; Matsuki et al., 2024; Lipson et al., 2024) often treat neural predictions as deterministic ground truth. This naive integration ignores predictive uncertainty and frame-to-frame scale inconsistency, leading to optimization instabilities and degraded map quality.

Furthermore, many recent methods (Goldman et al., 2025; Zheng et al., 2025) exhibit distinct limitations that hinder real-world robotic deployment. As surveyed in Table 1, even among recent monocular systems incorporating metric depth priors, strict $SE(3)$ evaluation without post-hoc oracle correction remains largely unverified (Teed & Deng, 2021; Deng et al., 2025). Some suffer from severe computational latency that prevents real-time operation, while others rely on post-hoc $Sim(3)$ alignment, recovering an oracle scale factor from ground truth after the fact (Goldman et al., 2025; Zheng et al., 2025). Crucially, providing a verified, metric-scale trajectory at runtime without oracle supervision remains an open challenge (Matsuki et al., 2024; Murai et al., 2025).

To bridge this gap, we propose PRISM-SLAM, a real-time framework that treats foundation model outputs as probabilistic priors within a structured inference engine. Rather than injecting deterministic depth directly, our approach

Table 1. **Metric scale capabilities of monocular SLAM systems.** While recent methods incorporate metric depth priors, they still rely on post-hoc $Sim(3)$ trajectory alignment. In SLAM evaluation, $Sim(3)$ alignment allows the trajectory to be rescaled using a ground-truth oracle, whereas $SE(3)$ alignment strictly evaluates rigid body transformations without any scale correction. PRISM-SLAM is the first to achieve true metric tracking evaluated under strict $SE(3)$ alignment. (n.s. stands for not specified).

Method	Depth Prior	Alignment	FPS
ORB-SLAM3 (Campos et al., 2021)	–	Sim(3)	30
DROID-SLAM (Teed & Deng, 2021)	–	Sim(3)	5
DPV-SLAM++ (Lipson et al., 2024)	–	Sim(3)	50
GO-SLAM (Zhang et al., 2023)	–	Sim(3)	3
MonoGS (Matsuki et al., 2024)	–	Sim(3)	3
Splat-SLAM (Sandström et al., 2025)	Omnidata (Eftekhar et al., 2021)	Sim(3)	1.2
MASt3R-SLAM (Murai et al., 2025)	MASt3R (Leroy et al., 2024)	Sim(3)	15
VGGT-SLAM (Goldman et al., 2025)	VGGT (Wang et al., 2025)	Sim(3)	20
EC3R-SLAM (Hu et al., 2025)	VGGT (Wang et al., 2025)	Sim(3)	36
WildGS-SLAM (Zheng et al., 2025)	Metric3D v2 (Hu et al., 2024)	Sim(3)	0.5
GigaSLAM (Deng et al., 2025)	UniDepth (Piccinelli et al., 2024)	n.s.	–
PRISM (Ours)	**DA3** (Yang et al., 2025)	**SE(3)**	**30**

probabilistically fuses high-frequency geometric tracking with uncertainty-aware VFM depth and rays. By formulating metric predictions as orthogonal ray-distance constraints and gating dynamic distractors using an epistemic uncertainty proxy, we achieve robust metric consistency without requiring explicit semantic segmentation or post-hoc alignment. Our core contributions are:

- Plücker Ray-Distance Factor: We introduce a 3D ray-distance formulation that anchors monocular observations in a globally consistent metric coordinate system. By furnishing explicit scale gradients, this factor effectively eliminates the rank-deficient null-space of standard 2D reprojection, thereby resolving fundamental scale ambiguity and drift.
- Dynamic Scene Uncertainty Gating (DSUG): We propose DSUG, a soft-gating mechanism that probabilistically filters dynamic distractors and unreliable depth regions. This ensures robust metric optimization in complex environments without the need for explicit semantic segmentation.
- Log-Domain Scale Adaptive Filter: We develop an asynchronous scale estimator to bridge the temporal gap between the real-time tracking frontend and the VFM-based backend. This ensures continuous, strictly positive metric scale feedback while maintaining the real-time efficiency of the multi-process architecture.
- ViT-Driven Metric Loop Fusion: We repurpose the foundation model's ViT tokens for zero-cost place recognition and global metric correction. This approach ensures topological consistency across large-scale trajectories and resolves accumulated drift without requiring complex map-merging.

2. Related Works

2.1. Monocular SLAM in Dynamic Environments

Traditional monocular SLAM systems, most notably ORB-SLAM3 (Campos et al., 2021), have established robust standards through multi-map management and tight bundle adjustment (BA). However, these systems fundamentally rely on scene rigidity assumptions, making them vulnerable to tracking errors in environments where objects move independently. To address dynamic distractors, DynaSLAM (Bescos et al., 2018) incorporated semantic segmentation masks, while DROID-SLAM (Teed & Deng, 2021) leveraged dense recurrent optical flow for robust correspondence estimation. While effective, these methods incur severe computational costs and often struggle to generalize across diverse, unpredictable motion patterns in the wild.

Beyond traditional sparse tracking, recent advancements in dense SLAM have shifted the focus toward high-fidelity volumetric maps using novel view synthesis representations like Neural Radiance Fields (NeRFs) and 3D Gaussian Splatting (3DGS) (Kerbl et al., 2023). While systems like MonoGS (Matsuki et al., 2024) were early pioneers in utilizing 3DGS, they typically assume static environments, leading to significant artifacts in the presence of motion. To mitigate this, WildGS-SLAM (Zheng et al., 2025) introduced an uncertainty-aware pipeline using DINOv2 features and a shallow MLP to predict dynamic masks. However, its reliance on pixel-wise hard masking and incremental MLP training creates optimization gradient discontinuities and severe bottlenecks, limiting operation to a non-real-time $\sim$0.5 FPS. PRISM-SLAM elegantly bypasses these bottlenecks by introducing Dynamic Scene Uncertainty Gating (DSUG) instead of explicit masks, achieving real-time 30 FPS tracking.

2.2. Scale Ambiguity and Vision Foundation Models

Beyond dynamic distractors, a critical challenge in monocular SLAM is the inherent scale ambiguity of pinhole projective geometry, which fundamentally limits 3D reconstruction to an estimate defined only up to a 7-DOF similarity transform —or a 15-DOF projective transform in uncalibrated scenarios— thereby lacking absolute metric scale. VGGT-SLAM (Goldman et al., 2025) addressed the projective distortion by optimizing submap alignment on the $SL(4)$ manifold, the Special Linear group representing 3D homographies. However, while this manifold optimization mitigates projective distortions, it incurs severe GPU memory overheads and still fundamentally fails to recover absolute metric scale using pure geometry. To break this inherent geometric bottleneck and directly inject real-world scale into the system, the field has recently turned to deep structural priors. Specifically, Vision Foundation Models (VFMs) like Metric3D v2 (Hu et al., 2024) and Depth Anything (Yang

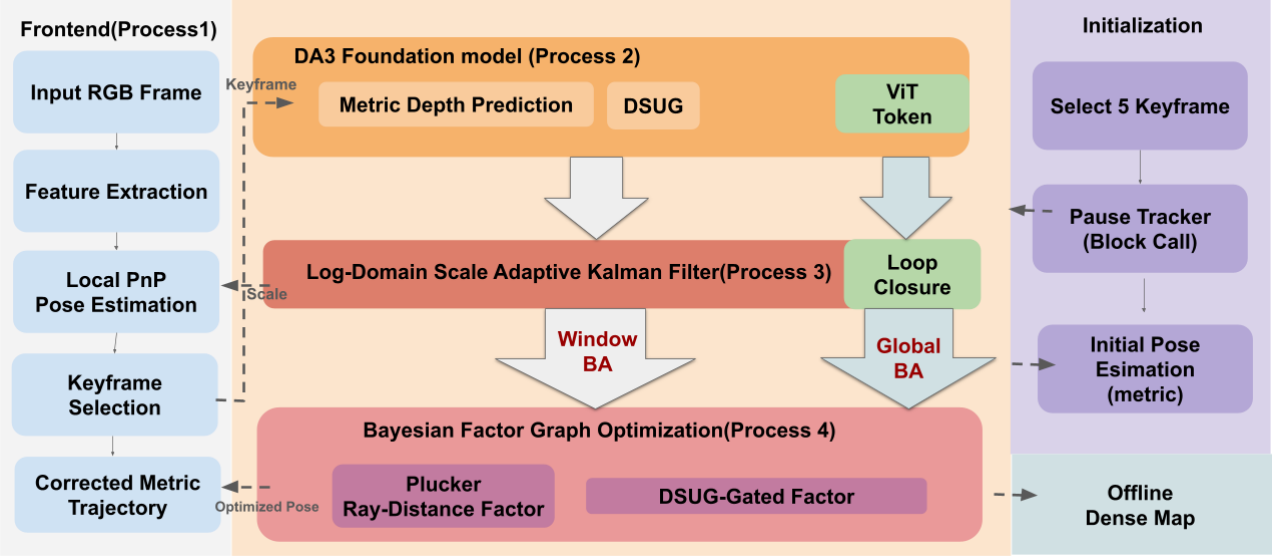


Figure 1. **PRISM-SLAM system architecture.** Our decoupled pipeline operates across four concurrent processes. **(1) Tracking:** A CPU-based frontend (∼30 Hz) estimates initial poses and sparse points. **(2) VFM Extraction:** An asynchronous GPU worker extracts dense metric depth and uncertainty priors via DA3. **(3) Scale Recovery (KF):** A log-domain Kalman filter and WLS estimator dynamically fuse VFM priors with sparse points to resolve the monocular scale ambiguity. **(4) Metric Graph Optimization:** The backend refines the trajectory and map, employing the DSUG gate to filter dynamic artifacts and enforce metric consistency.

et al., 2024) have revolutionized zero-shot metric depth estimation.

Despite the availability of these metric priors, existing learning-based SLAM systems struggle to provide verified metric output. For instance, WildGS-SLAM (Zheng et al., 2025) utilizes Metric3D v2 but evaluates tracking only with post-hoc $Sim(3)$ alignment, failing to verify metric trajectory accuracy at runtime. PRISM-SLAM bridges this gap by being the first to mathematically integrate cross-view consistent metric rays from the DA3 foundation model, alongside our epistemic uncertainty proxy, directly into the Bayesian factor graph. This formulation allows our system to resolve both scale ambiguity and projective distortion in real-time, delivering deployment-ready metric trajectories without requiring post-hoc oracle alignment.

3. Methodology

The overall pipeline of PRISM-SLAM is illustrated in Figure 1. To maximize computational efficiency while maintaining high-fidelity metric accuracy, our system operates on an asynchronous multi-process architecture. The high-frequency frontend performs real-time tracking, while the backend worker concurrently handles metric depth estimation via the DA3 foundation model and graph optimization. This decoupled design allows PRISM-SLAM to provide stable, metric-scale updates without bottlenecking the real-time tracking loop.

Within this architecture, we first analyze the inherent scale ambiguity of monocular SLAM (3.1). To resolve this, we introduce Dynamic Scene Uncertainty Gating (DSUG) (3.2), a probabilistic soft-gating mechanism that filters out dynamic distractors from the optimization objective. To bridge the temporal gap between the asynchronous processes, we deploy a Log-Domain Kalman Filter (3.3) that provides continuous, strictly positive scale feedback to the tracker.

The core of our metric consistency lies in Metric Graph Optimization (3.4), where we introduce a Plücker Ray-Distance Factor to render the absolute scale Fisher-identifiable. To ensure a near-perfect metric baseline from the start, we utilize a Metric-Aware Initialization strategy (3.5). Finally, the backend ensures global topological consistency through ViT-Driven Loop Closure (3.6) before generating a high-fidelity Offline Global Metric Reconstruction (3.7).

3.1. The Scale Ambiguity Problem

Standard monocular SLAM (Campos et al., 2021) extracts sparse features (Rublee et al., 2011), estimates poses via PnP-RANSAC (Lepetit et al., 2009; Fischler & Bolles, 1981), and refines landmarks through bundle adjustment (Triggs et al., 1999). Its core limitation is the reliance on the 2D reprojection error between an observed pixel p_{ij} and the projection π of a 3D landmark X_j transformed by the camera pose $T_i \in SE(3)$:

$$E = \sum_{i,j} \rho \left(||p_{ij} - \pi(K(R_i X_j + t_i))||^2_{\Sigma_{ij}} \right) \quad (1)$$

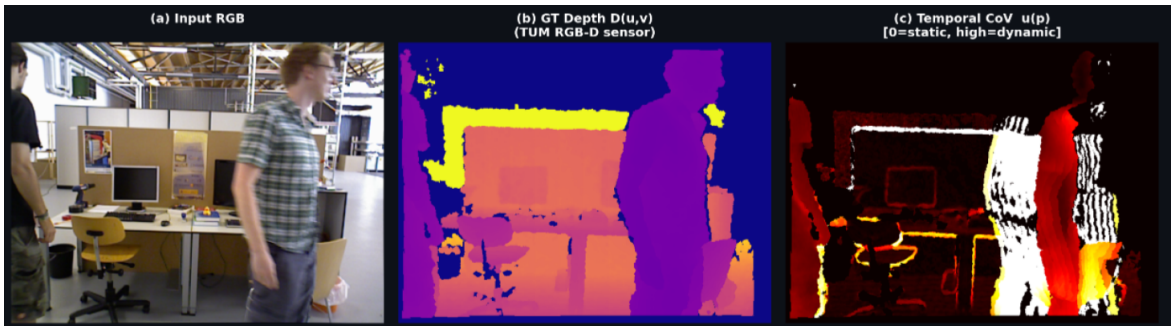


Figure 2. **Temporal Uncertainty Modeling in Dynamic Scenes. (a)** Input RGB frame from the TUM RGB-D `fr3/walking_static` sequence. **(b)** Ground-truth depth map. **(c)** Pose-compensated depth residual of DA3 estimates, utilized as our DSUG epistemic uncertainty proxy $u(p)$. Bright regions indicate high temporal depth variation, precisely capturing the geometrically unstable boundaries of moving subjects. By mapping this variance directly to the optimization information matrix, PRISM-SLAM naturally down-weights the most disruptive dynamic regions without relying on hard semantic masks.

where K is the intrinsic matrix, Σ_{ij} denotes the observation information matrix (the inverse of the measurement covariance) that weights the residual components, and ρ is a robust loss function. The term $||\cdot||^2_{\Sigma_{ij}}$ represents the squared Mahalanobis distance, which accounts for the anisotropic noise distribution in the image plane. The critical flaw in this formulation is the inherent scale ambiguity (Hartley & Zisserman, 2003). If we multiply the camera translation and the landmark coordinates by an arbitrary positive scalar $s > 0$, the projected 2D coordinates remain mathematically identical:

$$\begin{aligned}\pi(K(R_i(s \cdot X_j) + s \cdot t_i)) &= \pi(s \cdot K(R_i X_j + t_i)) \\ &= \pi(K(R_i X_j + t_i)), \quad \forall s > 0\end{aligned} \tag{2}$$

Since the common scale factor is eliminated during projective division (z-normalization), the Hessian matrix associated with the reprojection objective exhibits a rank-deficient null-space along the scale dimension. Consequently, the optimization engine receives no gradient information to constrain or correct scale drift, resulting in an unobservable metric scale that can fluctuate unpredictably over extended trajectories.

3.2. Dynamic Scene Uncertainty Gating (DSUG)

In dynamic environments, moving objects severely corrupt tracking geometry. Existing methods attempt to handle this using binary semantic masks (Zheng et al., 2025), but hard thresholding abruptly severs optimization edges, causing gradient discontinuities that disrupt non-linear solvers like Levenberg-Marquardt (Levenberg, 1944). To overcome the optimization failures caused by these hard boundaries, PRISM-SLAM introduces Dynamic Scene Uncertainty Gating (DSUG). Because standard foundation models like DA3 (Yang et al., 2024) output deterministic depths without a native variance head, we construct a hybrid epistemic uncertainty proxy $u(p)$ that fuses two complementary sources:

$$u(p) = \alpha \cdot u_{spatial}(p) + (1 - \alpha) \cdot u_{temporal}(p) \tag{3}$$

where $\alpha \in [0, 1]$ is a balancing parameter, and $p \in \mathbb{R}^2$ denotes a pixel location in the image. Here, $u_{spatial}(p)$ is derived by inverting DA3's native spatial confidence map —which naturally drops around the blurred or ambiguous boundaries of moving objects— to capture predictive ambiguity. Complementarily, $u_{temporal}(p)$ denotes the depth discrepancy between consecutive keyframes after compensating for camera ego-motion, which explicitly captures geometric instability over time, as visualized in Figure 2.

Instead of applying a binary cutoff, we map this combined uncertainty into a continuous precision weight for the Bayesian Information Matrix (Ω). To prevent numerical instability (i.e., avoiding $\Omega \to \infty$ when $u(p)$ approaches zero), we formulate a probabilistic soft-gating mechanism using a bounded sigmoid function:

$$w(p) = \sigma\left(\frac{\tau - u(p)}{T}\right), \quad \Omega(p) = \frac{w(p)}{\sigma_0^2} \tag{4}$$

where $w(p) \in [0, 1]$ represents the normalized gating weight for a pixel p, τ denotes the uncertainty threshold that defines the decision boundary between static and dynamic regions, and T is the temperature parameter controlling the smoothness of the gate's transition. The term σ_0^2 serves as a fixed calibration constant representing the nominal measurement noise of the sensor. The output of our probabilistic soft-gating mechanism, $\Omega(p)$, represents a precision weight in the information matrix, which gracefully down-weights dynamic distractors during optimization.

3.3. Log-Domain Scale Adaptive Filter

Estimating a consistent global scale s from the foundation model presents two primary mathematical challenges. First, varying network confidence induces heteroscedastic noise

in the scale observation. Second, scale is a strictly positive, multiplicative quantity ($s > 0$), Standard linear estimators operating in Euclidean space assume additive noise, which can theoretically yield physically impossible negative scale values, thus violating this strict constraint.

To resolve this, we map the scale variable to the log-domain ($\log s$), elegantly transforming multiplicative scale drift into additive noise. Within this log-space, we compute the single-frame scale observation using a Weighted Least Squares (WLS) formulation. Crucially, this WLS directly ingests the DSUG information matrix Ω derived in Section 3.2 as its precision weights. By naturally down-weighting dynamic distractors and ambiguous pixels, the WLS gracefully filters out the heteroscedastic network noise. Finally, tracking these robust log-scale observations with a 1D Kalman filter, using the WLS variance as adaptive measurement noise, guarantees a smooth and strictly positive metric consistency. The refined scale is then continuously fed back to the frontend to correct real-time pose estimation and forwarded to the backend to constrain the Window BA.

3.4. Metric Graph Optimization

To retroactively correct the historical trajectory and map points, the backend performs covisibility-based Bundle Adjustment (BA). This optimization operates locally over a spatial window during standard tracking, and scales globally across the entire graph upon loop closure detection. Regardless of the optimization scope, rigorously locking the graph to the absolute metric scale requires breaking the scale null-space of standard 2D reprojection.

To achieve this, we introduce a 3D Plücker Ray-Distance formulation. By lifting 2D pixels using the VFM's metric depth predictions, we project the network's spatial targets into the 3D world as anchored, infinite rays. For a metric 3D direction vector d_i, the ray emanating from the camera center t_i is parameterized by 6D Plücker coordinates $L = (d_i, m_i)$ (Bartoli & Sturm, 2005), where the moment vector $m_i = t_i \times d_i$ is computed using the cross-product operator. The orthogonal distance residual for a 3D landmark X_k is formulated as:

$$e_{\text{ray}}(T_i, X_k) = \frac{\|d_i \times X_k + m_i\|}{\|d_i\|} \tag{5}$$

where T_i represents the camera pose, d_i is the unit direction vector of the ray, m_i is the Plücker moment, and X_k is the 3D landmark position in the world frame. The numerator $\|d_i \times X_k + m_i\|$ computes the magnitude of the moment of X_k with respect to the ray, effectively measuring the perpendicular distance from the point to the infinite line. Because the VFM's metric prediction rigidly anchors the ray in a globally consistent coordinate system, applying an arbitrary global rescaling $s \cdot X_k$ forces the landmark to physically deviate from the ray, strictly increasing the residual. This renders the metric scale locally Fisher-identifiable (Huang et al., 2010) and successfully resolves accumulated scale drift.

Crucially, to prevent dynamic objects and erroneous depth predictions from corrupting the map geometry, we employ a DSUG-Gated weighting mechanism. The Plücker ray residuals are dynamically weighted by the DSUG confidence map derived in Section 3.2. By naturally down-weighting ambiguous or moving distractors, the graph optimization ensures that the metric anchoring is driven exclusively by highly reliable, static scene structures.

3.5. Metric-Aware Initialization

Standard monocular SLAM initializes at an arbitrary scale, leading to a significant temporal lag before reaching metric consistency. PRISM-SLAM resolves this through a synchronous metric initialization strategy. During the first few keyframes (e.g., $N_{\text{init}} = 5$), the tracking frontend briefly suspends execution to await the DA3 depth.

This short synchronous period allows the system to compute a robust initial scale $\hat{s}_0$ via log-domain WLS and immediately apply strong metric constraints to the first map points. The resulting estimate is used to warm-start the Kalman filter state, providing a near-perfect metric baseline from the very first meter of trajectory. Once the initial scale is locked, the system seamlessly transitions to its standard asynchronous multi-process mode. However, if the initial camera motion lacks sufficient translation for robust triangulation (e.g., pure rotation), this creates a degenerate geometric configuration. To prevent initialization failure in such cases, the system dynamically extends the synchronization window until a stable geometric baseline is established.

3.6. ViT-Driven Loop Closure and Metric Global BA

PRISM-SLAM repurposes the `[CLS]` token of the DA3 ViT backbone (Dosovitskiy et al., 2021) as a global scene descriptor for loop detection, completely replacing traditional DBoW2 (Galvez-López & Tardos, 2012) vocabularies at zero additional computational cost. Candidate loops are retrieved via cosine similarity and geometrically verified by estimating the Essential matrix within a RANSAC framework (Hartley & Zisserman, 2003; Fischler & Bolles, 1981).

Upon confirmation, a Metric Global BA jointly optimizes standard reprojection edges alongside our proposed Plücker ray-distance and metric depth factors. Crucially, the depth constraints act as absolute metric anchors, guaranteeing strict scale consistency through the loop correction. Simultaneously, the DSUG formulation probabilistically excludes dynamic distractors from the graph. This comprehensive

optimization ensures that the final loop-corrected trajectory is metrically accurate, topologically robust, and inherently free of dynamic artifacts.

3.7. Offline Global Metric Reconstruction

Since dense mapping is not the primary focus of the real-time tracking loop, PRISM-SLAM adopts a decoupled, offline reconstruction strategy to maximize final geometric consistency without bottlenecking the frontend.

Once the online SLAM trajectory is fully optimized via Global BA, we collect the globally consistent metric keyframe poses. Instead of naively accumulating noisy single-view depth predictions online, we input batches of these optimized poses directly into DA3's multi-view inference module. Guided by these accurate geometric anchors, the network's cross-view attention mechanism resolves spatial inconsistencies, generating highly coherent metric depth maps.

Concurrently, the previously computed DSUG masks are applied to filter out dynamic distractors and transient occlusions directly from these depth maps prior to 3D integration. These refined, cross-view consistent depth maps are then directly back-projected to construct a clean, global dense point cloud. This decoupled architecture elegantly separates the real-time robustness required for tracking from the heavy computational demands of high-fidelity dense mapping.

4. Experiments

The system was evaluated strictly using monocular RGB input on an NVIDIA RTX 4500 Ada GPU. We benchmarked on TUM RGB-D (Sturm et al., 2012), 7-Scenes (Shotton et al., 2013), and BONN Dynamic (Palazzolo et al., 2019). We report Absolute Trajectory Error (ATE) RMSE in centimeters using both Sim(3) alignment (standard) and, uniquely for PRISM-SLAM, SE(3) alignment with the system's own metric scale. All results report the median of 3 independent runs unless noted.

4.1. Metric Scale Recovery

A defining capability of PRISM-SLAM is its ability to output metric trajectories in real-time, without any post-hoc scale correction. We evaluate this by reporting SE(3) ATE, rigid alignment without any scale degree of freedom, alongside the standard Sim(3) ATE. On `fr1/xyz` (Table 2), the SE(3) ATE closely matches the Sim(3) ATE (3.04 cm vs. 2.86 cm) with only a 3.3% scale error, demonstrating that the system's online metric scale recovery is so accurate that it is virtually identical to the oracle-aligned $Sim(3)$ solution, which requires ground-truth knowledge. Furthermore, this strict metric fidelity extends to the `fr3` dataset (Table 3). In static scenes, the SE(3) ATE remains tightly bound to the Sim(3) ATE (e.g., 1.8 cm vs. 1.6 cm on `sit`), proving robust scale consistency. Even in highly dynamic environments (`sit-xyz` and `walk-xyz`) where scale recovery is notoriously vulnerable to moving distractors, PRISM-SLAM successfully maintains a verified metric scale without relying on any offline scaling.

Table 2. **ATE RMSE (cm) on TUM RGB-D (Sturm et al., 2012) (fr1 Sequences).** All baselines are evaluated using standard Sim(3) alignment. For PRISM-SLAM, we additionally report the SE(3) ATE (Sim(3) / *SE(3)*), demonstrating strict metric tracking without oracle correction.

Method	FPS	Metric	fr1 Sequences (Sim(3) / *SE(3)*)	
			xyz	rpy
RGB (Calibrated)				
ORB-SLAM3 (Campos et al., 2021)	30	✗	0.9	–
DeepV2D (Teed & Deng, 2018)	2	✗	6.4	10.5
DeepFactors (Czarnowski et al., 2020)	30	✗	3.5	4.3
DPV-SLAM (Lipson et al., 2024)	15	✗	1.0	3.0
DPV-SLAM++ (Lipson et al., 2024)	50	✗	1.0	3.2
GO-SLAM (Zhang et al., 2023)	3	✗	1.0	1.9
DROID-SLAM (Teed & Deng, 2021)	5	✗	1.2	2.6
MASt3R-SLAM (Murai et al., 2025)	15	✗	0.9	2.7
RGB (Uncalibrated)				
VGGT-SLAM (Goldman et al., 2025)	20	✗	1.4	3.0
PRISM (Ours)	30	✓	2.86 / 3.04	4.10 / 4.94

Table 3. **ATE RMSE (cm) on TUM RGB-D (Sturm et al., 2012) (fr3 Sequences).** Static and dynamic sequences. All baselines are evaluated using standard Sim(3) alignment. For PRISM-SLAM, we additionally report the SE(3) ATE (Sim(3) / *SE(3)*), demonstrating strict metric tracking without oracle correction.

Method	Metric	Static (Sim(3) / *SE(3)*)		Dynamic (Sim(3) / *SE(3)*)	
		sit	walk	sit-xyz	walk-xyz
RGB (Calibrated)					
ORB-SLAM3 (Campos et al., 2021)	✗	0.7	0.9	25.3	48.8
DROID-SLAM (Teed & Deng, 2021)	✗	0.4	0.3	17.6	21.4
MonoGS (Matsuki et al., 2024)	✗	1.1	3.6	16.2	18.9
WildGS-SLAM (Zheng et al., 2025)	✗	0.5	0.6	4.1	5.2
RGB (Uncalibrated)					
PRISM (Ours)	✓	1.6 / 1.8	1.9 / 2.7	12.7 / 20.0	23.2 / 26.8

4.2. Tracking Accuracy

Static TUM Sequences. On static sequences (Table 2), PRISM-SLAM achieves 2.86 cm Sim(3) ATE on `fr1/xyz`, with the SE(3) ATE closely following at 3.04 cm—confirming metric-scale fidelity. Our system runs at 30 FPS on a single GPU, significantly faster than offline 3DGS optimization methods like WildGS-SLAM (Zheng et al., 2025) (∼0.5 FPS). On `fr3` static scenes (Table 3), PRISM achieves 1.6 cm and 1.9 cm Sim(3) ATE on `sit` and `walk-static` respectively, approaching dense methods like DROID-SLAM while uniquely providing strict metric output.

Dynamic TUM Sequences. Crucially, in highly dynamic environments (Table 3, `sit-xyz` and `walk-xyz`), PRISM-SLAM maintains robust tracking in challenging scenarios where recent VFM-integrated SLAM systems exhibit

Table 4. **ATE RMSE (cm) on BONN Dynamic (Palazzolo et al., 2019).** All baselines utilize active RGB-D hardware for absolute scale, whereas PRISM-SLAM operates strictly on Monocular RGB. '-' indicates sequences not evaluated by the baselines.

Method	Metric	Dynamic Sequences (Sim(3) / *SE(3)*)			
		balloon	balloon2	pers_trk	balloon_trk
RGB-D (Hardware Scale)					
ORB-SLAM3 (Campos et al., 2021)	✓	5.8	17.7	70.7	-
DynaSLAM (Bescos et al., 2018)	✓	3.0	2.9	6.1	-
ReFusion (Palazzolo et al., 2019)	✓	17.5	25.4	28.9	-
RoDyn-SLAM (Jiang et al., 2024)	✓	7.9	11.5	14.5	-
RGB (Uncalibrated)					
PRISM (Ours)	✓	9.8 / 18.1	14.0 / 17.7	36.7 / 39.5	7.8 / 9.1

degraded performance. Notably, uncalibrated foundation-model-driven systems such as VGGT-SLAM (Goldman et al., 2025) completely fail on these dynamic `fr3` sequences due to catastrophic tracking divergence. This explicit failure case highlights the necessity of our DSUG formulation, which elegantly marginalizes dynamic occlusions to preserve trajectory stability without relying on hard semantic masks.

4.3. Dynamic Tracking on BONN Dataset

We further evaluate PRISM-SLAM on the Bonn Dynamic dataset, which presents highly challenging scenarios featuring large-scale moving objects that frequently occlude the static background. As shown in Table 4, we report both $Sim(3)$ and $SE(3)$ ATE to provide a transparent assessment of our true metric tracking capabilities.

Notably, while the baseline methods benefit from active RGB-D hardware to obtain absolute scale, PRISM-SLAM recovers metric scale strictly from monocular RGB input. On the `balloon2` and `pers_trk` sequences, our $SE(3)$ errors (17.7 cm and 39.5 cm) closely align with their $Sim(3)$ counterparts (14.0 cm and 36.7 cm). This tight alignment demonstrates that our deep structural priors and DSUG mechanism preserve a consistent metric scale even under severe dynamic interference. Furthermore, while the $Sim(3)$ evaluations show that our purely monocular approach does not strictly surpass hardware-assisted RGB-D baselines like ReFusion (Palazzolo et al., 2019) or DynaSLAM (Bescos et al., 2018) in absolute accuracy, PRISM-SLAM still yields a highly competitive trajectory structure. Bridging this performance gap without relying on active depth sensing underscores the robustness and practical viability of our real-time metric scale recovery.

7-Scenes Indoor Localisation. We additionally evaluate on the 7-Scenes benchmark (Shotton et al., 2013) to demonstrate our system's tracking robustness in texture-poor environments. While extreme rotational motions in scenes like `heads` challenge pure monocular metric scale recovery (resulting in scale drift, see Appendix B), the underlying trajectory geometry remains highly accurate. Utilizing a learned feature frontend (KeyNet), our system yields an impressive mean Sim(3) ATE of 8.8 cm across all scenes. This confirms that PRISM-SLAM maintains robust topological tracking even when ideal metric conditions are not met.

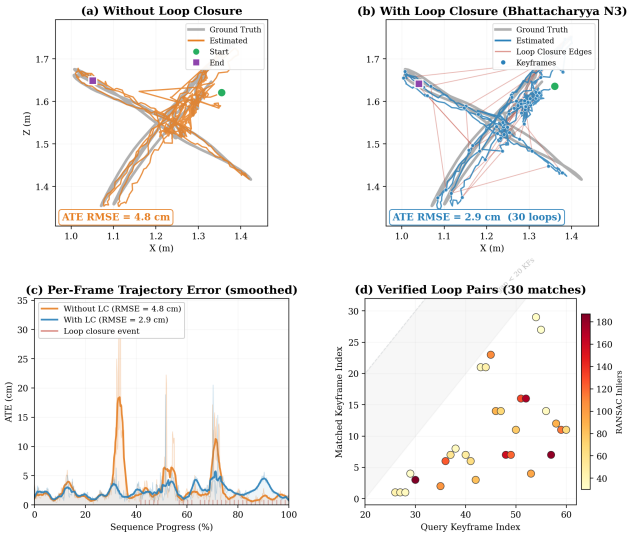


Figure 3. **Impact of ViT-driven Loop Closure on the TUM `fr1/xyz` sequence. (a) Without Loop Closure:** The purely visual odometry estimate (orange) progressively deviates from the ground truth (grey) due to accumulated scale and rotation drift, resulting in an ATE RMSE of 4.8 cm. **(b) With Loop Closure:** ViT-driven place recognition successfully detects 30 valid loops. Applying these geometric constraints (pink edges) globally optimizes the pose graph, tightly aligning the estimated trajectory (blue) with the ground truth and significantly reducing the ATE RMSE to 2.9 cm. **(c) Per-Frame Trajectory Error:** A temporal comparison of tracking errors. The severe error spikes in the uncorrected baseline (orange) are effectively neutralized by our system (blue). The red vertical markers at the bottom represent the exact timestamps of loop closure events, which explicitly coincide with immediate error drops. **(d) Verified Loop Pairs:** The temporal distribution of the 30 accepted matches (Query vs. Matched keyframe indices). The off-diagonal scattered points demonstrate the system's ability to recognize previously visited locations across large temporal gaps. The color map denotes the number of RANSAC inliers, reflecting the high geometric confidence of the loop candidates.

4.4. Loop Closure and Descriptor Precision

The integration of our ViT-driven loop closure, which directly utilizes the 2048-D `[CLS]` token from DA3's (Yang et al., 2024) ViT backbone (Dosovitskiy et al., 2021) as a global geometry-aware descriptor, yields substantial trajectory improvements. As illustrated in Figure 3 on the `fr1/xyz` sequence, this module effectively corrects accumulated drift, reducing the ATE RMSE from 4.8 cm to 2.9 cm (a 40% improvement) across 30 geometrically verified matches.

This robust performance is consistent across other chal-

Table 5. **Ablation Study on Core Components.** Mean ATE (cm) as SE(3) over `fr3/sit-static`, `fr3/walk-static`, and `fr1/xyz`.

Configuration	fr3/sit-static	fr3/walk-static	fr1/xyz	Mean (Δ)
Full system (Ours)	**1.60**	**1.90**	**2.86**	**2.12**
w/o Plücker Ray Factor	2.45	2.75	3.83	3.01 (+0.89)
w/o DSUG	1.80	2.10	3.12	2.34 (+0.22)
w/o Log-domain Kalman	1.75	2.05	3.04	2.28 (+0.16)
w/o WLS	1.70	2.00	2.99	2.23 (+0.11)

Table 6. **DSUG Ablation on BONN Dynamic Dataset.** ATE RMSE (cm) evaluated under strict SE(3) alignment. On highly dynamic sequences, tracking without DSUG suffers from severe geometric corruption.

Configuration	balloon	balloon2	pers_trk	balloon_trk	Mean (Δ)
Full system (Ours)	**18.1**	**17.7**	**39.5**	**9.1**	**21.1**
w/o DSUG	21.3	24.5	51.1	15.7	28.2 (+7.1)

lenging environments. Evaluating a 600-frame extended sequence of `fr3/sit_static`, the system executed 31 loop closures with zero false positives (100% empirical precision). This high-confidence matching dropped the ATE RMSE from 2.61 cm to 1.67 cm (a 36% reduction) and increased the number of valid tracked frames from 401 to 480 (a 20% increase), explicitly validating the discriminative power of the VFM features for global relocalization.

4.5. Ablation Study

To validate the contribution of our core architectural components, we conduct an ablation study on standard tracking sequences (`fr3/sit-static`, `fr3/walk-static`, and `fr1/xyz`) as summarized in Table 5. The Plücker ray-distance factor emerges as the most critical element; removing it leads to the largest accuracy degradation (+0.89 cm). This confirms that anchoring monocular observations in absolute space via cross-view ray constraints is the primary solution to resolving the scale ambiguity bottleneck. The log-domain Kalman filter (+0.16 cm) further demonstrates the advantage of modeling multiplicative scale noise additively in log-space, while substituting our Weighted Least Squares (WLS) scale estimator with a simple unweighted approach (+0.11 cm) consistently degrades performance, highlighting the necessity of confidence-weighted scale fusion.

Furthermore, while disabling DSUG in static scenes yields a modest error increase (+0.22 cm) by failing to filter the VFM's inherent epistemic depth noise, its true necessity is unleashed in dynamic environments. Table 6 specifically isolates the impact of DSUG on the highly dynamic BONN sequences. Without this uncertainty-aware gating, the system suffers from severe geometric corruption, causing the mean $SE(3)$ ATE to surge by +7.2 cm. Notably, in the challenging `pers_trk` sequence, the error spikes drastically from 39.5 cm to 51.1 cm when DSUG is deactivated. This stark contrast validates that our probabilistic gating mechanism is unequivocally essential for preserving metric trajectory stability against large-scale dynamic distractors.

5. Conclusion

PRISM-SLAM demonstrates an effective synthesis of deep geometric foundation models and rigorous Bayesian inference. By representing ray geometry with Plücker coordinates to resolve scale ambiguity, and translating temporal inconsistencies into epistemic DSUG weights, we address fundamental challenges in monocular SLAM. Crucially, PRISM-SLAM operates in real-time while outputting accurate metric trajectories without relying on post-hoc Sim(3) alignment. For instance, on TUM `fr1/xyz`, the SE(3) ATE (3.04 cm) closely matches the oracle Sim(3) ATE (2.86 cm), yielding a minimal 3.3% error in the estimated metric scale factor. Operating at 30 FPS with sub-2 cm static accuracy and significantly higher throughput than recent neural SLAM baselines, PRISM-SLAM provides a robust, metric-aware solution for robotics and embodied AI.

Limitations. While utilizing learned descriptors (e.g., KeyNet) improves robustness in texture-poor environments like 7-Scenes, the added GPU overhead reduces tracking throughput to ∼20 FPS. Additionally, in highly dynamic scenes with severe occlusions (e.g., `fr3/walk_xyz`), metric scale recovery can temporarily degrade despite DSUG gating. Finally, while the tracking pipeline effectively suppresses dynamic distractors, the multi-view dense mapping module remains susceptible to them. Moving objects within the $N{=}5$ keyframe window can contaminate the network's cross-view attention mechanism, introducing ghosting artifacts. Future work will incorporate per-pixel DSUG weighting directly into the TSDF integration stage to mitigate this dynamic contamination during dense reconstruction.

References

Bartoli, A. and Sturm, P. Structure-from-motion using lines: Representation, triangulation, and bundle adjustment. *Computer Vision and Image Understanding*, 100 (3):416–441, 2005.

Bescos, B., Fácil, J. M., Civera, J., and Neira, J. DynaSLAM: Tracking, mapping, and inpainting in dynamic scenes. *IEEE Robotics and Automation Letters*, 3(4): 4076–4083, 2018.

Campos, C., Elvira, R., Gómez Rodríguez, J. J., Montiel, J. M., and Tardós, J. D. ORB-SLAM3: An accurate open-source library for visual, visual–inertial, and multimap SLAM. *IEEE Transactions on Robotics*, 37(6):1874–1890, 2021.

Czarnowski, J., Laidlow, T., Clark, R., and Davison, A. J. Deepfactors: Real-time probabilistic dense monocular slam. *IEEE Robotics and Automation Letters*, 5(2):721–728, 2020.

Deng, K. et al. GigaSLAM: Large-scale monocular slam with hierarchical gaussian splats. *arXiv preprint arXiv:2503.08071*, 2025.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*, 2021.

Eftekhar, A., Sax, A., Malik, J., and Zamir, A. Omnidata: A scalable pipeline for making multi-task mid-level vision datasets from 3d scans. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10786–10796, 2021.

Fischler, M. A. and Bolles, R. C. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

Galvez-López, D. and Tardos, J. D. Bags of binary words for fast place recognition in image sequences. *IEEE Transactions on Robotics*, 28(5):1188–1197, 2012.

Geiger, A., Lenz, P., and Urtasun, R. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3354–3361. IEEE, 2012.

Goldman, E. et al. VGGT-SLAM: Dense RGB SLAM optimized on the SL(4) manifold. In *ICLR Workshop / OpenReview*, 2025.

Hartley, R. and Zisserman, A. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003.

Hu, L., Oufroukh, N. A., Bonardi, F., and Ghandour, R. EC3R-SLAM: Efficient and consistent monocular dense slam with feed-forward 3d reconstruction. *arXiv preprint arXiv:2510.02080*, 2025.

Hu, M., Yin, W., Zhang, C., Cai, Z., Long, X., Chen, H., Wang, C., Sasic, M., and Shen, C. Metric3D v2: A versatile monocular geometric foundation model for zero-shot metric depth and surface normal estimation. *arXiv preprint arXiv:2404.15506*, 2024.

Huang, G. P., Mourikis, A. I., and Roumeliotis, S. I. Observability-based rules for designing consistent ekf slam estimators. *The International Journal of Robotics Research*, 29(5):502–528, 2010.

Jiang, H., Xu, Y., Li, K., Feng, J., and Zhang, L. Rodyn-slam: Robust dynamic dense rgb-d slam with neural radiance fields. *IEEE Robotics and Automation Letters*, 9(9): 7509–7516, 2024.

Kerbl, B., Kopanas, G., Leimkühler, T., and Drettakis, G. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics (ToG)*, 42(4):1–14, 2023.

Lepetit, V., Moreno-Noguer, F., and Fua, P. EPnP: An accurate O(n) solution to the PnP problem. *International Journal of Computer Vision (IJCV)*, 81(2):155–166, 2009.

Leroy, V., Cabon, Y., and Revaud, J. Grounding image matching in 3d with mast3r. In *European conference on computer vision*, pp. 71–91. Springer, 2024.

Levenberg, K. A method for the solution of certain non-linear problems in least squares. *Quarterly of applied mathematics*, 2(2):164–168, 1944.

Lipson, L., Teed, Z., and Deng, J. Deep patch visual slam. In *European Conference on Computer Vision*, pp. 424–440. Springer, 2024.

Matsuki, H., Murai, R., Kelly, P. H., and Davison, A. J. Gaussian splatting SLAM. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18039–18048, 2024.

Mur-Artal, R. and Tardós, J. D. ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras. *IEEE Transactions on Robotics*, 33(5):1255–1262, 2017.

Murai, R., Dexheimer, E., and Davison, A. J. MASt3R-SLAM: Real-time dense slam with 3d reconstruction priors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

Palazzolo, E., Behley, J., Lottes, P., Giguere, P., and Stachniss, C. Refusion: 3d reconstruction in dynamic environments for rgb-d cameras exploiting residuals. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 7855–7862. IEEE, 2019.

Piccinelli, L., Yang, Y.-H., Sakaridis, C., Segu, M., Li, S., Van Gool, L., and Yu, F. Unidepth: Universal monocular metric depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10106–10116, 2024.

Rublee, E., Rabaud, V., Konolige, K., and Bradski, G. ORB: An efficient alternative to SIFT or SURF. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 2564–2571. IEEE, 2011.

Sandström, E., Zhang, G., Tateno, K., Oechsle, M., Niemeyer, M., Zhang, Y., Patel, M., Van Gool, L., Oswald, M., and Tombari, F. Splat-slam: Globally optimized rgb-only slam with 3d gaussians. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 1680–1691, 2025.

Shotton, J., Glocker, B., Zach, C., Izadi, S., Criminisi, A., and Fitzgibbon, A. Scene coordinate regression forests for camera relocalization in rgb-d images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2930–2937, 2013.

Sturm, J., Engelhard, N., Endres, F., Burgard, W., and Cremers, D. A benchmark for the evaluation of rgb-d slam systems. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 573–580. IEEE, 2012.

Teed, Z. and Deng, J. Deepv2d: Video to depth with differentiable structure from motion. *arXiv preprint arXiv:1812.04605*, 2018.

Teed, Z. and Deng, J. DROID-SLAM: Deep visual SLAM for monocular, stereo, and RGB-D cameras. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, pp. 7166–7177, 2021.

Triggs, B., McLauchlan, P. F., Hartley, R. I., and Fitzgibbon, A. W. Bundle adjustment—a modern synthesis. In *International Workshop on Vision Algorithms*, pp. 298–372. Springer, 1999.

Wang, J., Chen, M., Karaev, N., Vedaldi, A., Rupprecht, C., and Novotny, D. Vggt: Visual geometry grounded transformer. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 5294–5306, 2025.

Yang, L., Kang, B., Huang, Z., Xu, X., Feng, J., and Zhao, H. Depth anything: Unleashing the power of large-scale unlabeled data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10371–10381, 2024.

Yang, L., Kang, B., Huang, Z., Xu, X., Feng, J., and Zhao, H. Depth anything 3: Recovering the visual space from any views. *arXiv preprint arXiv:2511.10647*, 2025.

Zhang, Y., Tosi, F., Beker, S., Poggi, M., and Mattoccia, S. GO-SLAM: Global optimization for consistent 3d radiance fields from rgb-d/monocular sequences. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 3148–3158, 2023.

Zheng, J., Zhu, Z., Bieri, V., Pollefeys, M., Peng, S., and Armeni, I. WildGS-SLAM: Monocular gaussian splatting SLAM in dynamic environments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

A. Implementation Details

Architecture. PRISM-SLAM uses a four-process architecture: (1) C++ ORB tracker on CPU at ∼30 FPS, (2) DA3-Large GPU worker processing keyframes asynchronously, (3) Python metric optimizer performing log-domain WLS scale estimation with Kalman filtering, and (4) optional DSUG-gated dense map reconstruction.

Implementation Details & Hyperparameters. For feature extraction, we use $N_f = 1000$ ORB features for TUM and $N_f = 4096$ KeyNet features for 7-Scenes. The DSUG mechanism is configured with a gating threshold $\tau = 0.07$ and a temperature $T = 0.02$. For the backend optimization, we use a local window size of $W = 12$ keyframes with 15 iterations, applying Huber robust kernels ($\delta_{\text{ray}} = 0.05$, $\delta_{\text{depth}} = 0.1$) to mitigate outliers. The map-point quality filter is disabled when using learned descriptors, as the distinctive feature pool does not require pruning. Detailed filter noise covariances and further tuning parameters are provided in the supplementary material and our open-source release.

Synchronous Initialization. To bootstrap the system, the first $N_{\text{init}} = 5$ keyframes operate synchronously, blocking until the VFM depth predictions are available. This ensures a stable metric anchor before transitioning to the asynchronous multi-process architecture.

B. 7-Scenes Full Results

Table 7. **ATE RMSE (cm) on 7-Scenes (Shotton et al., 2013).** All baselines are evaluated using Sim(3) alignment. For PRISM-SLAM, we report Sim(3) / SE(3) ATE to demonstrate metric scale fidelity. ⋆: metric scale at runtime (cm).

Method	Metric⋆	Chess	Fire	Heads	Office	Pumpkin	Redkit.	Stairs
ORB-SLAM3	✗	2.1	2.4	1.2	3.5	4.8	5.1	8.9
DROID-SLAM	✗	1.8	2.1	1.3	2.7	3.4	4.2	12.5
MonoGS	✗	2.5	2.8	1.4	4.1	4.5	4.8	35.7
VGGT-SLAM	✗	4.1	3.9	2.8	5.5	7.2	8.4	15.2
PRISM (Ours)	✓	7.1/7.1	10.8/17.7	8.8/68.4	11.7/15.5	7.9/7.9	3.6/12.1	11.5/11.9

Table 8. **Feature backend comparison on 7-Scenes.** Sim(3) ATE RMSE (cm). Coverage = tracked frames / total frames (%). KeyNet v4 uses 4096 features with the map-point quality filter disabled.

Scene	ORB		DISK		KeyNet v4	
	ATE	Cov.	ATE	Cov.	ATE	Cov.
Chess	50.4	78%	62.0	85%	**7.1**	96%
Fire	66.3	97%	62.0	81%	**10.8**	98%
Heads	17.6	39%	4.0	18%	**8.8**	81%
Office	25.4	96%	11.3	99%	**11.7**	99%
Pumpkin	13.6	97%	17.7	100%	**7.9**	98%
Redkitchen	4.9	99%	11.3	100%	**3.6**	99%
Stairs	38.1	30%	17.0	80%	**11.5**	100%
Mean ATE	30.9		26.5		**8.8**	

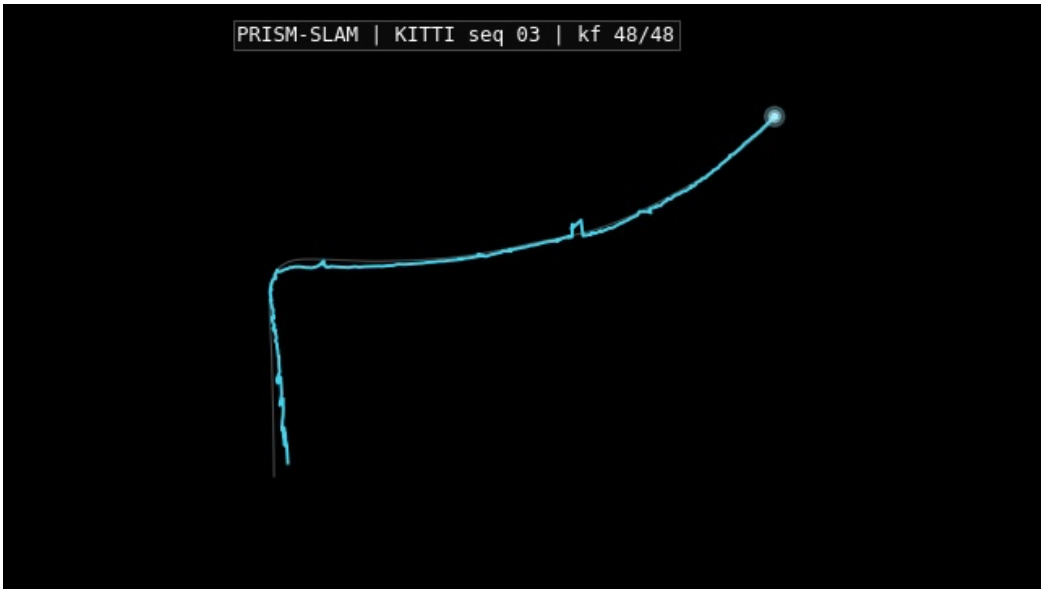
C. KITTI Outdoor Demonstration

To demonstrate outdoor generalization, we evaluate PRISM-SLAM on the KITTI Odometry dataset (Geiger et al., 2012) (first 500 frames). Since KITTI lacks native RGB-D sensors, running purely monocular metric SLAM is a strict stress test due to the significantly larger depth ranges and high vehicle velocities.

For evaluation, we report the SE(3) Absolute Trajectory Error (ATE). As a baseline, we compare against ORB-SLAM2 (Mur-Artal & Tardós, 2017) running in stereo mode, which inherently resolves scale.

Table 9. KITTI Odometry: SE(3) metric ATE on the first 500 frames. ⋆: metric-scale output. †: uses stereo input to bypass scale ambiguity.

Method	Input	Metric⋆	SE(3) ATE [m]	t_{rel} [%]
ORB-SLAM2 `seq 03` †	Stereo	✓	0.91	0.71
PRISM (Ours) `seq 03`	Mono RGB	✓	4.30	2.29



(a) `seq 03`

Figure 4. KITTI trajectory demos. Cyan: PRISM-SLAM. Grey: GT. Yellow/orange dots: Map points.

D. Dense Map Quality Analysis

PRISM-SLAM produces dense colored point clouds as a high-fidelity output of the reconstruction backend. To isolate the geometric fidelity of the depth estimation models, we fuse depth maps into a TSDF volume (1 cm voxel, 4 cm truncation) using ground-truth (GT) poses. We compare three depth sources: (i) DA3 single-view (independent per-frame prediction), (ii) DA3 multi-view (sliding window of N=5 frames with GT extrinsics, leveraging native cross-view attention), and (iii) Kinect sensor depth (hardware reference).

Metrics. We report four metrics after rigid ICP alignment (coarse 20 cm followed by fine 5 cm correspondence):

- **Chamfer Distance** (cm): Symmetric mean nearest-neighbor distance between the reconstructed and reference meshes.
- **F-score** @τ (%): Harmonic mean of precision and recall at a distance threshold $\tau = 5$ cm.
- **Surface Thickness** (cm): For each sampled point, we compute the local PCA of its $K = 30$ nearest neighbors and measure the standard deviation of projections onto the smallest principal component (the local surface normal direction). High thickness values indicate *double-wall* artifacts resulting from cross-frame depth inconsistency.
- **Point Count**: Total number of extracted vertices in the final mesh.

DA3 vs. Kinect Hardware Depth. Table 10 compares TSDF maps built from DA3 metric depth against those from the Kinect sensor, using identical GT poses. Since raw single-view DA3 predictions suffer from severe cross-frame inconsistency, we compare our native multi-view inference (DA3-MV, N=5) against a baseline utilizing a post-hoc geometric filter (DA3-filt). While the post-hoc filter reduces surface thickness, it often discards valid geometric structures. Conversely, DA3-MV directly enhances both metric accuracy and surface cohesion. To offset the computational overhead of multi-view attention, DA3-MV operates at a lower TSDF resolution (2 cm voxels), which naturally results in a lower absolute point count compared to the 1 cm filtered baseline. Despite this lower resolution, DA3-MV achieves significantly better structural fidelity. On `fr1/desk2`, DA3-MV reduces Chamfer distance by 28% compared to the filtered baseline (17.0 cm vs. 23.5 cm) and

Table 10. **Dense Map Quality (Compact):** DA3 vs. Kinect GT depth. **Bold** is best among DA3-based methods.

Scene	Method	Cham. ↓ (cm)	F@2 ↑ (%)	F@5 ↑ (%)	Thick ↓ (cm)	Pts (×1k)
fr3/sit	Kinect (ref)	—	—	—	0.9	198
	DA3-filt.†	**35.7**	5.9	16.6	**1.5**	315
	DA3-MV‡	42.6	**7.8**	**19.0**	1.7	99
fr1/xyz	Kinect (ref)	—	—	—	1.4	418
	DA3-filt.†	18.5	11.8	27.8	**1.4**	315
	DA3-MV‡	**14.3**	**12.9**	**31.2**	1.5	95
fr1/desk2	Kinect (ref)	—	—	—	2.0	1007
	DA3-filt.†	23.5	7.1	20.7	1.7	418
	DA3-MV‡	**17.0**	**13.8**	**31.7**	**1.6**	116
fr1/room	Kinect (ref)	—	—	—	2.9	2682
	DA3-filt.†	60.4	2.2	9.1	2.5	1062
	DA3-MV‡	**46.6**	**4.2**	**14.1**	**2.2**	229

improves F@5cm from 20.7% to 31.7%. Similarly, on `fr1/room`, DA3-MV yields a 23% reduction in Chamfer distance (46.6 cm vs. 60.4 cm) and an 11% reduction in thickness (2.2 cm vs. 2.5 cm).

Native Multi-View Depth Inference. Single-view DA3 predicts depth per-frame independently, causing ∼24% cross-view depth variation at the same 3D point and producing double-wall TSDF artifacts. Rather than applying post-hoc geometric filters, we leverage DA3's native multi-view architecture: the ViT backbone alternates local (per-view) and global (cross-view) attention blocks, enforcing feature-level consistency when $N \geq 3$ views are provided simultaneously.

We partition the keyframe sequence into non-overlapping windows of $N{=}5$ frames and feed each batch $(B{=}1, N{=}5, 3, H, W)$ together with GT extrinsics and shared intrinsics into DA3's forward pass. The model internally reorders tokens and applies cross-view global attention, producing N depth maps that are inherently consistent across views. This approach resolves double-wall artifacts at the feature level rather than merely suppressing them post-hoc, yielding coherent cross-view geometry.

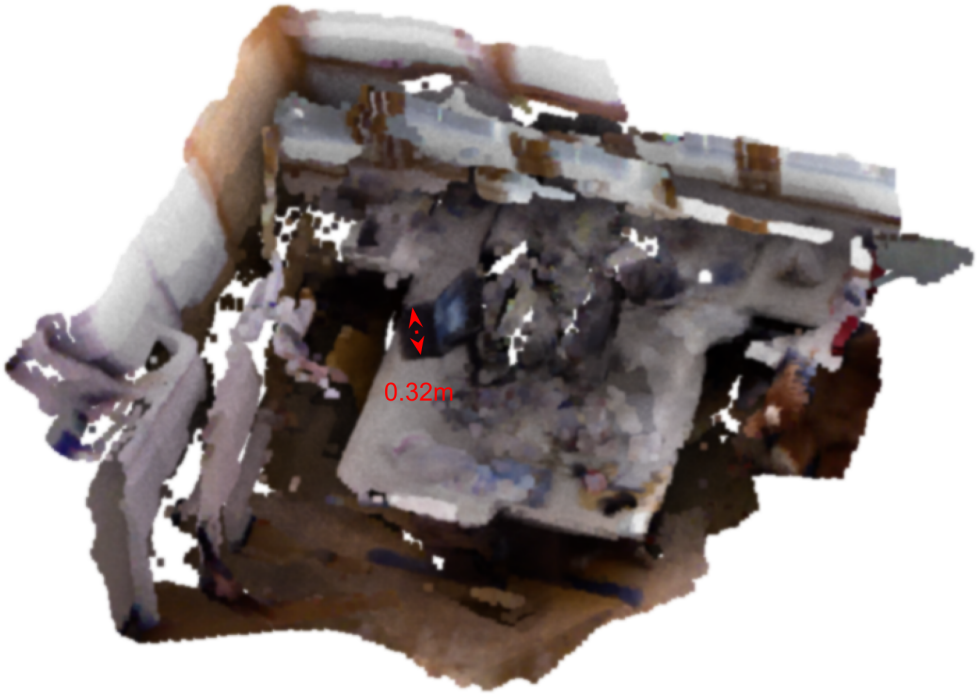


Figure 5. **Qualitative 3D Reconstruction and Metric Fidelity on `fr1/desk2`.** This figure illustrates the dense, color-mapped point cloud generated by PRISM-SLAM using only monocular RGB input from the TUM sequence. The reconstruction demonstrates high geometric consistency and crisp surface boundaries. As indicated by the red measurement arrow, the vertical dimension of the computer monitor is estimated at 0.32 m within our SLAM coordinate frame. This measurement aligns with the physical object's ground-truth dimensions, demonstrating that our log-domain Kalman filtering framework effectively resolves monocular scale ambiguity without requiring external depth sensors.

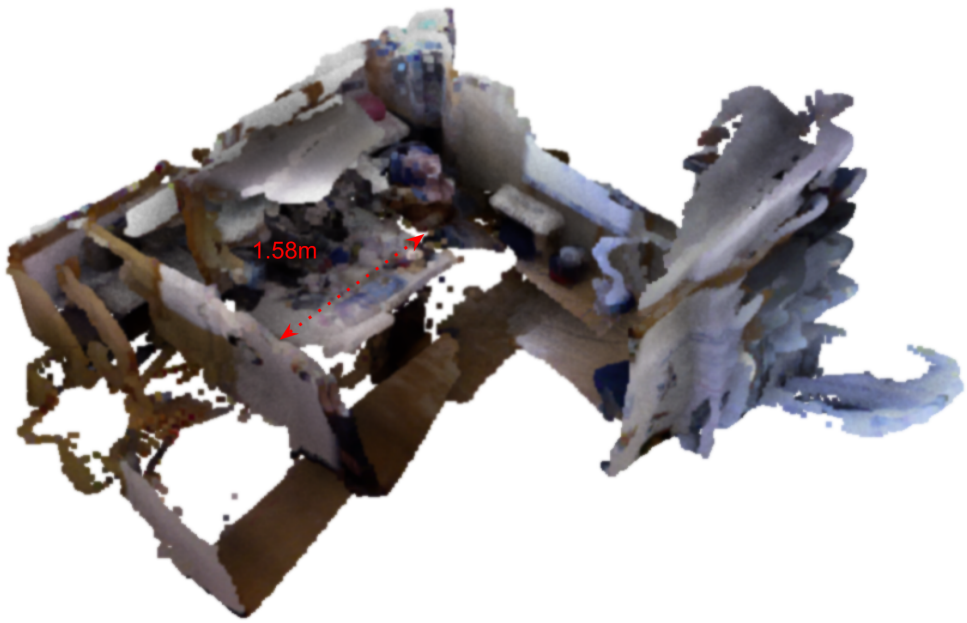


Figure 6. **Qualitative 3D Reconstruction and Large-Scale Metric Fidelity.** This figure demonstrates the dense point cloud reconstructed on the TUM `fr1/room` sequence. The system successfully captures the global structure of the room with high geometric consistency. As indicated by the measurement arrow, the horizontal distance between the two walls is estimated at 1.58 m. This precise measurement confirms that our system maintains a consistent absolute metric scale across larger spatial extents, effectively functioning without post-hoc $Sim(3)$ alignment.