HIP3R: HIERARCHICAL POSE GRAPHS FOR LONG-HORIZON 3D RECONSTRUCTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Feed-forward models accurately estimate camera motion and 3D geometry from short video windows. However, computational and memory costs prevent direct inference on sequences containing thousands of frames. Long videos must therefore be processed as overlapping chunks. This division produces independent local reconstructions with inconsistent coordinate frames and duplicate estimates of shared cameras. Achieving global consistency requires propagating constraints from adjacent overlaps and distant revisits across the sequence. This requirement naturally motivates pose graph optimization. However, the choice of graph level creates a trade-off. *Chunk-level* graphs efficiently resolve scale drift and close large loops. Yet, their rigid transformations cannot correct internal errors or conflicting frame predictions. Conversely, *frame-level* optimization offers local flexibility, but scaling it globally is computationally expensive and can break metric scale. HIP3R resolves this trade-off without retraining the underlying models, using a hierarchical pose graph. HIP3R first optimizes a robust chunk-level $\mathrm{Sim}(3)$ graph to establish the global layout and scale. It then holds this layout fixed and estimates one shared $\mathrm{SE}(3)$ camera per frame, turning overlapping window predictions into a single trajectory. As a post-processing method that does not retrain the backend, HIP3R applies to both chunk-based and streaming models. Across seven benchmarks, HIP3R reduces VGGT-Omega average trajectory error by 74%, from 18.555 to 4.906 m, and relative tracking drift by 94%, from 0.9878 to 0.0622 m. It also decreases 3D surface reconstruction error by 64%, from 0.0759 to 0.0272 m. When paired with LingBot-MAP, HIP3R reduces trajectory error by 52%, from 8.288 to 4.017 m, and achieves the lowest overall error. It executes faster than existing alignment methods on long sequences.

1 INTRODUCTION

Recovering globally consistent 3D geometry and camera trajectories from video is fundamental to computer vision. This capability supports autonomous navigation, robotic mapping, and spatial artificial intelligence. Recent 3D foundation models (Wang et al., 2024; 2025a; 2026b) have significantly advanced visual geometry. By jointly predicting camera poses and dense pointmaps in a single forward pass, they bypass classical handcrafted matching. These models achieve high accuracy on short video clips without per-scene calibration or optimization. However, the quadratic complexity of visual attention limits them to short temporal windows. Directly scaling transformers to extended multi-kilometer sequences remains too expensive.

To process long videos, modern systems adopt a divide-and-conquer strategy (Deng et al., 2025). Long sequences are divided into overlapping video chunks, which are processed independently and stitched together. While chunking bounds GPU memory, sequential alignment introduces severe computational and numerical bottlenecks. Dense overlapping points share high spatial redundancy, making all-pixel alignment computationally expensive. Furthermore, pairwise alignment errors compound along the sequential chain, causing severe scale and orientation drift. Loop closures are essential to eliminate cumulative drift and enforce global topological consistency. However, when the chains are not yet aligned, applying multiple long-range loop constraints at once can make the non-convex optimization unstable. The solver may diverge or converge to poor local minima.

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.

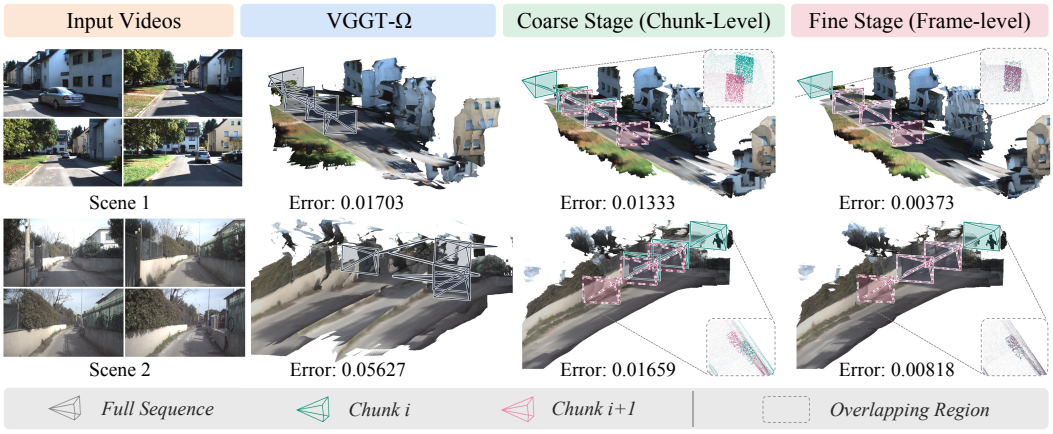


Figure 1: Overlapping predictions recover local geometry in inconsistent frames. Coarse layout optimization corrects global placement and scale, but keeps each chunk's internal cameras. Shared-camera trajectory refinement estimates one trajectory from predictions and within-chunk motion.

As shown in Figure 1, even after global chunk placement is resolved, a conflict remains at the frame level. Existing alignment methods (Deng et al., 2025; Zhang et al., 2026b; Xiong et al., 2026) optimize similarity transformations per chunk. However, a single similarity transform preserves internal camera configurations up to scale. Consequently, it cannot resolve cross-window camera-pose inconsistency. Overlapping chunks produce multiple, conflicting pose estimates of the same frame under differing visual contexts. A chunk is an inference unit, whereas the true trajectory contains only one camera pose per frame. Arbitrarily selecting one prediction discards multi-view evidence and causes velocity jumps across chunk boundaries. Conversely, full bundle adjustment is too expensive, while relative motion alone drifts from the recovered global layout. Standard pose graph optimization lacks a shared camera representation for duplicate observations.

To overcome these challenges, we introduce HIP3R, a Hierarchical Pose-graph optimization method for long-horizon 3D Reconstruction. HIP3R decomposes reconstruction into *coarse chunk* layout recovery and *fine frame* trajectory refinement. The coarse stage optimizes a *chunk-level* $\mathrm{Sim}(3)$ pose graph over adjacent overlaps and distant loop closures. This stage establishes a globally consistent layout and eliminates accumulated scale drift. Holding this coarse layout fixed, the fine stage optimizes a *frame-level* shared-camera $\mathrm{SE}(3)$ pose graph, representing each physical frame with a single camera variable to resolve cross-window pose inconsistency. Placement priors keep that camera on the recovered layout, while within-chunk motion factors resolve conflicting predictions from overlapping windows. The result is one trajectory, not a stack of rigidly transformed chunks. We evaluate HIP3R across seven public benchmarks spanning diverse indoor and outdoor environments. Under matched backends and windows, HIP3R reduces VBR trajectory error from 41.27 to 22.44 m relative to loop-enabled VGGT-Long. Controlled ablations confirm that bounded point sampling cuts graph build and solve time by 83.62% at unchanged accuracy. Stage-wise results show that the two optimization stages play different roles. Additional evaluations show that the method also works with streaming backends such as LingBot-MAP. We summarize our main contributions as follows:

- We show that chunk-level similarity alignment alone cannot ensure cross-window camera-pose consistency, as overlapping windows may yield conflicting estimates for the same frame.
- We develop a hierarchical pose graph to recover globally consistent chunk layout and scale, followed by a shared $\mathrm{SE}(3)$ camera pose for each frame.
- We introduce progressive loop closure and bounded point sampling to achieve stable and scalable optimization on long sequences without retraining.

2 Related Work

Feed-forward 3D vision models. Feed-forward 3D models have emerged as an alternative to classical Structure-from-Motion (SfM). Pioneering architectures like DUSt3R (Wang et al., 2024) and

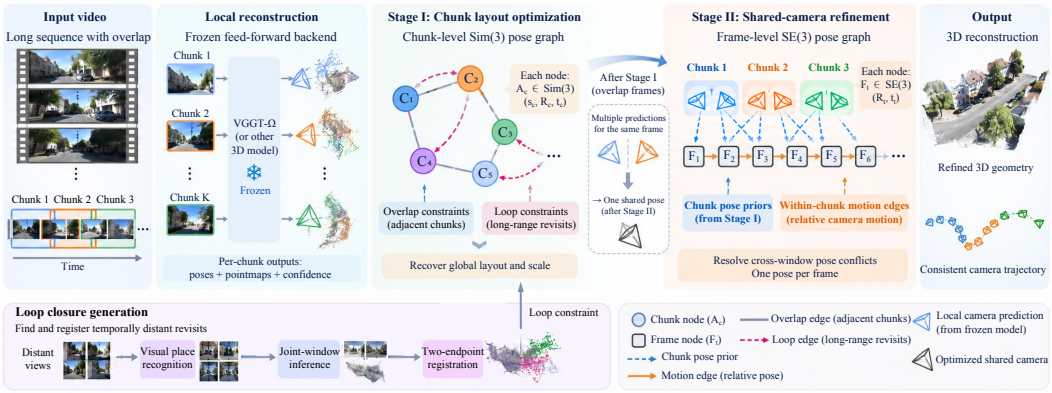


Figure 2: Overview of HIP3R. Frozen backends predict overlapping chunks. Stage I builds a chunk-level $\mathrm{Sim}(3)$ graph from overlap and loop constraints to recover layout and scale. Stage II estimates one shared $\mathrm{SE}(3)$ camera per frame under that layout.

MASt3R (Leroy et al., 2024) directly regress dense pointmaps from uncalibrated image pairs. Recent multi-view transformers such as VGGT (Wang et al., 2025a) and VGGT-Omega (Wang et al., 2026b) extend feed-forward predictions across broader view collections. Subsequent models explore permutation invariance (Wang et al., 2025c), sparse attention (Wang et al., 2026a), and learned scene actions (Wang & Agapito, 2026). These models achieve high local geometric accuracy without per-scene numerical optimization. However, quadratic attention restricts them to short windows, so long sequences yield overlapping local reconstructions.

Streaming and recurrent video geometry. To extend temporal context beyond static windows, another line of research develops streaming and recurrent architectures. Early methods such as Spann3R (Wang & Agapito, 2025) and CUT3R (Wang et al., 2025b) maintain persistent spatial memory states. Recent methods like StreamVGGT (Zhuo et al., 2026), STream3R (Lan et al., 2026), and WinT3R (Li et al., 2026) process video frames causally with temporal key-value caching. LingBot-MAP (Chen et al., 2026a) structures geometric context across anchor frames, local windows, and trajectory memory. Other approaches explore test-time training, memory compression, and feature retrieval (Chen et al., 2026b; Zhang et al., 2026a; Xie et al., 2026; Jin et al., 2026; Liu et al., 2026; Zou et al., 2026). Causal streaming still accumulates drift on long trajectories. As post-processing, HIP3R adds global loops and per-frame pose consistency without retraining.

Multi-window fusion and Structure-from-Motion. Scaling feed-forward predictions to extended trajectories connects to classical global Structure-from-Motion and pose graph optimization. Chunk-based methods like VGGT-Long (Deng et al., 2025), VGGT-Align (Zhang et al., 2026b), and VGGT-Motion (Xiong et al., 2026) align adjacent chunks via pairwise $\mathrm{Sim}(3)$ alignment. A rigid $\mathrm{Sim}(3)$ per chunk preserves internal cameras up to scale and cannot resolve overlapping-frame disagreements. GlueMap (Pan et al., 2026) couples feed-forward star reconstructions with global motion averaging and bundle adjustment over virtual tracks. Track-level bundle adjustment is expensive, and standard pose graphs lack a shared camera for duplicate observations. HIP3R recovers chunk-level $\mathrm{Sim}(3)$ layout, then estimates one shared $\mathrm{SE}(3)$ camera per frame.

3 METHOD

3.1 PROBLEM FORMULATION

Let $\mathcal{I} = \{I_t\}_{t=1}^{T}$ denote an input video sequence of T frames. We partition $\mathcal{I}$ into K overlapping chunks $\{C_k\}_{k=1}^{K}$ of length C with overlap O. Each chunk is processed independently by a feed-forward 3D vision model (Wang et al., 2026b). For chunk C_k, the backend predicts dense pointmaps $X_{k,i}$, confidence maps $c_{k,i}$, and camera-to-chunk poses $P_{k,i}$. Here $i \in \{1, \ldots, C\}$ indexes local frames within the chunk. Let $g(k, i) \in \{1, \ldots, T\}$ map local frame i in chunk C_k to its global

video index. The goal is to recover a single camera trajectory $\mathcal{F} = \{F_t \in \mathrm{SE}(3)\}_{t=1}^{T}$ together with consistent 3D pointmaps. Figure 2 gives an overview.

Decoupled optimization. Directly optimizing all camera and point variables across thousands of frames is expensive. Unconstrained bundle adjustment loses the efficiency of feed-forward models. HIP3R uses a two-level hierarchical pose graph. Stage I solves a chunk-level $\mathrm{Sim}(3)$ pose graph for a globally aligned layout (Section 3.2). Stage II then optimizes a frame-level $\mathrm{SE}(3)$ pose graph with one shared camera variable per frame, with that layout held (Section 3.3).

3.2 Coarse Layout Optimization with Progressive Loop Scheduling

We formulate global layout recovery as optimization over a similarity pose graph $\mathcal{G}_c = (\mathcal{V}_c, \mathcal{E}_c)$. Each node $k \in \mathcal{V}_c$ represents a chunk C_k and is associated with a similarity transformation $A_k = (s_k, R_k, t_k) \in \mathrm{Sim}(3)$. This transformation maps local chunk coordinates to a common world coordinate system as $A_k(x) = s_k R_k x + t_k$. An edge $(i, j) \in \mathcal{E}_c$ represents a measured relative transformation $\widehat{Z}_{ij}$ from the coordinate system of chunk i to that of chunk j. The corresponding predicted relative transformation is $A_j^{-1} A_i$. Overlap edges connect consecutive chunks, while loop edges connect non-consecutive chunks that observe the same physical scene. Relative constraints determine scale only up to a global factor, so we fix A_1 as the world frame and scale reference. Scale errors in independent chunk predictions accumulate on long trajectories, so we optimize the remaining s_k jointly with rotation and translation.

Dense overlapping 3D points are spatially redundant, so all-pixel alignment is unnecessary. For adjacent chunks, identical pixels in overlapping frames yield paired 3D predictions. We estimate relative similarities with confidence-weighted IRLS rather than unweighted Umeyama on all dense pixels (Deng et al., 2025). Each iteration runs a weighted Umeyama alignment (Umeyama, 1991) and updates Huber weights from spatial residuals. We sample at most 2×10^5 pairs evenly from the flattened arrays, excluding invalid points and non-positive confidence.

For non-adjacent chunks, visual place recognition proposes candidate loop closures across wide temporal separations (Izquierdo & Civera, 2024). Following coarse-to-fine loop filtering, candidate loop pairs are registered at both endpoints. For each selected candidate pair, we construct a joint input from surrounding frames and infer local geometry via the backend. Aligning the joint prediction back to original chunk coordinates yields the relative loop measurement $\widehat{Z}_{ij} = B_j B_i^{-1}$. The alignment fit statistic q_{ij} is the IRLS residual p_{95}, falling back to p_{90} when p_{95} is unavailable. It maps to a bounded weight $w_{ij} = \mathrm{clip}((1 + \max(q_{ij}, 0)/\sigma)^{-1}, w_{\min}, w_{\max})$ with σ=0.05, $w_{\min}$=0.05, and $w_{\max}$=1. This downweights inaccurate loop alignments based on geometric registration consistency. Combining sequential overlap and loop constraints, we define the objective over active edges $\mathcal{E}_c$. Let $E_{ij}^c = \widehat{Z}_{ij}^{-1} A_j^{-1} A_i$ denote the relative similarity error. We define the residual vector as

$$r_{ij}^c = \big(\mathrm{Log}_{\mathrm{SO}(3)} R(E_{ij}^c),\, \tau(E_{ij}^c),\, \log s(E_{ij}^c)\big) \in \mathbb{R}^7, \tag{1}$$

where τ extracts Euclidean translation error, and $\log s$ evaluates log-scale error. Because feed-forward backends predict geometry without analytical pose covariances, w_{ij} provides a data-driven confidence weight per edge. The macro layout objective minimizes the robust loss

$$A^\star = \arg\min_A \sum_{(i,j)\in\mathcal{E}_c} \sum_{\ell=1}^{7} \rho_\delta\big(\sqrt{w_{ij}}\, r_{ij,\ell}^c\big)\,, \tag{2}$$

where ρ_δ denotes the Huber penalty with scale $\delta = 1$, applied coordinate-wise to suppress large outliers. The trust-region solver uses adaptive Jacobi scaling to maintain comparable step magnitudes.

Activating all distant loops on an unaligned chain makes the non-convex solve unstable. We initialize chunk placements from sequential alignments and sort loop edges by temporal span. Shorter-span loops enter first over four progressive passes, retaining adjacent edges and warm-starting later stages. When constraints cluster around dense revisit hubs, the solver adds continuation stages and resets bounds before the full graph. Each free chunk has an absolute rotation vector, translation, and log-scale. We solve Eq. (2) with bounded trust-region reflective least squares and adaptive Jacobi scaling, limiting log-scale changes relative to initialization. Each residual depends on two chunks, so we form float64 endpoint derivatives by automatic differentiation and assemble a sparse Jacobian.

3.3 SHARED-CAMERA TRAJECTORY REFINEMENT

Given the recovered layout $A^\star$, overlapping frames still have multiple conflicting pose estimates. Selecting one prediction discards alternative observations and causes velocity jumps across chunk boundaries. We therefore introduce a shared camera-to-world variable $F_t \in \mathrm{SE}(3)$ for each frame t. We represent each camera pose $P_{k,i}$ by its rotation $R_{k,i} \in \mathrm{SO}(3)$ and camera center $c_{k,i} \in \mathbb{R}^3$, writing $P_{k,i} = [R_{k,i} \mid c_{k,i}]$. Its placement $Q_{k,i}$ in the world frame is

$$R(Q_{k,i}) = R_k R_{k,i}, \qquad c(Q_{k,i}) = s_k R_k c_{k,i} + t_k, \tag{3}$$

with $A_k = A_k^\star$. The chunk scale applies to camera centers. The shared pose is rigid with no scale variable. Soft placement priors pull cameras toward this reference while allowing adjustments.

As shown in Figure 2, the camera graph uses two types of geometric factors. For consecutive retained frames i, j in chunk k, define the relative motion measurement $M_{k,i\to j} = Q_{k,j}^{-1} Q_{k,i}$. With $E^f = M_{k,i\to j}^{-1} F_{g(k,j)}^{-1} F_{g(k,i)}$, the relative motion edge residual is

$$r_{k,i,j}^f = \big(\mathrm{Log}_{\mathrm{SO}(3)} R(E^f),\, \tau(E^f)\big) \in \mathbb{R}^6. \tag{4}$$

Here τ extracts Euclidean translation, and the rotation residual uses the standard rotation logarithm (Solà et al., 2018). Each placed prediction also provides a unary chunk pose prior on its shared camera, with $E^p = Q_{k,i}^{-1} F_{g(k,i)}$ and

$$r_{k,i}^p = \big(\mathrm{Log}_{\mathrm{SO}(3)} R(E^p),\, \tau(E^p)\big) \in \mathbb{R}^6. \tag{5}$$

When a physical frame appears in multiple overlapping chunks, their respective chunk pose priors constrain the same shared variable F_t. Relative motion edges connected to this node join local chunk trajectories across boundaries. The shared camera variable assigns one pose to each frame, while priors regularize conflicting predictions. From a maximum a posteriori perspective, the fine objective balances inter-frame observation likelihoods against global placement priors,

$$F^\star = \arg\min_F \sum_{(k,i,j)\in\mathcal{E}_f} \sum_{\ell=1}^{6} \rho_\delta\Big(\sqrt{\lambda_f}\, r_{k,i,j,\ell}^f(A^\star, F)\Big) + \sum_{(k,i)\in\mathcal{E}_p} \sum_{\ell=1}^{6} \rho_\delta\Big(\sqrt{\lambda_p}\, r_{k,i,\ell}^p(A^\star, F)\Big). \tag{6}$$

Consecutive frames share dense visual overlap and have little relative drift, so motion factors use $\lambda_f = 1.0$. Placement priors use $\lambda_p = 0.25$ as lower-frequency anchors to the recovered layout. The first camera is pinned to its coarse placement as the coordinate gauge. Closed-form Jacobian blocks are given in Appendix B. For extended sequences exceeding 2000 frames, we use a keyframe subgraph for efficiency. We retain chunk endpoints, shared overlap frames, and regularly spaced keyframes. Motion edges connect consecutive retained keyframes within each chunk. For coarse pose Q_t, define its correction as $\Delta_t = F_t Q_t^{-1}$. We interpolate corrections between keyframes and set $F_t = \Delta_t Q_t$. Appendix C details keyframe selection and interpolation. For geometry, points in chunk coordinates are placed by $A_k^\star$ and follow the shared cameras via $F_{g(k,i)} Q_{k,i}^{-1}$. The maps keep the coarse scale and are observed by one camera per frame.

4 EXPERIMENTS

We evaluate trajectory accuracy across seven public benchmarks comprising 61 diverse sequences. The benchmark spans indoor room-scale environments, including 7-Scenes (Shotton et al., 2013) and Neural RGB-D (Azinović et al., 2022). It also includes high-precision multi-scale scenes from ETH3D (Schöps et al., 2017) and Tanks and Temples (Knapitsch et al., 2017). Finally, it includes extended outdoor driving and campus traversals from Oxford Spires (Tao et al., 2025), KITTI (Geiger et al., 2012), and VBR (Brizi et al., 2024). Dense surface geometry is evaluated on ETH3D, Neural RGB-D, and 7-Scenes, which provide millimeter-accurate laser-scanned ground truth. Appendix E details dataset compositions and sequence splits. Baselines in Table 1 use released checkpoints and recommended window policies on identical sequences. VGGT-Long and VGGT-Align are evaluated from public implementations, while VGGT-Motion (Xiong et al., 2026) is not publicly available.

We evaluate trajectory accuracy using ATE, $\mathrm{RPE}_{\mathrm{rot}}$, and $\mathrm{RPE}_{\mathrm{trans}}$ (lower is better). ATE is translation RMSE after a per-sequence $\mathrm{Sim}(3)$ alignment with evo (`correct_scale`). Relative errors use all

consecutive frame pairs (Δ=1 frame). Rotation RPE is in degrees, and translation RPE applies the same $\mathrm{Sim}(3)$ correction. Pairwise pose AUC (higher is better) evaluates all $i < j$ camera pairs at angular thresholds of $5°$, $15°$, and $30°$. Reconstructed 3D pointmaps are evaluated using the mean of accuracy and completeness (Chamfer, meters) after Umeyama and ICP, with F1 at a 0.25 m hit distance. Appendix H provides complete per-dataset geometric breakdowns.

We integrate HIP3R with VGGT-Omega (Wang et al., 2026b) and LingBot-MAP (Chen et al., 2026a) through a unified pose and pointmap interface, unprojecting depth when needed. Both backends share the same graph without fine-tuning. As post-processing, HIP3R adds loop-window inference for revisits after local predictions. LingBot-MAP uses the $120/32$ windowed interface.

For hyperparameters, fine residual weights are set to $\lambda_f = 1.0$ and $\lambda_p = 0.25$ with Huber scale $\delta = 1.0$. The coarse stage solves the $\mathrm{Sim}(3)$ chunk graph using bounded trust-region reflective least squares with adaptive Jacobi scaling. Termination tolerances are set to $\texttt{ftol} = \texttt{xtol} = \texttt{gtol} = 10^{-6}$. The coarse optimizer is allocated up to 5000 function evaluations. The fine stage optimizes the $\mathrm{SE}(3)$ camera graph with up to 100 evaluations at tolerance 10^{-8}. For sequences with $T \leq 2000$, all physical frames remain active optimization variables. Longer sequences ($T > 2000$) adopt a stride-5 keyframe subgraph with interpolated residual corrections. Appendices B and D detail Jacobian constructions and solver settings. The default backend is VGGT-Omega 1B.

4.1 Benchmark Evaluation across Public Datasets

Table 1 reports macro-averaged tracking over all 61 sequences. With VGGT-Omega, HIP3R reduces ATE by 74%, from 18.555 to 4.906 m, $\mathrm{RPE}_{\mathrm{rot}}$ from $1.820°$ to $0.590°$, and $\mathrm{RPE}_{\mathrm{trans}}$ by 94%, from 0.9878 to 0.0622 m. On long traversals, sequential stitching accumulates scale and orientation drift. Progressive loop scheduling in the coarse layout stage (Section 3.2) corrects global drift and drives much of the ATE reduction in Table 1. Shared-camera refinement (Section 3.3) further improves relative pose accuracy while preserving the layout. On compact indoor benchmarks such as 7-Scenes and Neural RGB-D, native VGGT-Omega already achieves ATE of 0.0370 m and 0.0190 m, leaving limited room for further global correction. With LingBot-MAP, HIP3R reduces macro ATE by 52%, from 8.288 to 4.017 m, without retraining.

Table 1: Trajectory evaluation across indoor and outdoor benchmarks. Values are dataset-macro averages of ATE (m), $\mathrm{RPE}_{\mathrm{rot}}$ (°), and $\mathrm{RPE}_{\mathrm{trans}}$ (m). Bold marks the best result.

Method	**Mean ATE** ↓	**Mean $\mathrm{RPE}_{\mathrm{rot}}$** ↓	**Mean $\mathrm{RPE}_{\mathrm{trans}}$** ↓
CUT3R (Wang et al., 2025b)	44.894	4.469	1.4600
TTT3R (Chen et al., 2026b)	23.775	5.077	0.8077
LoGeR (Zhang et al., 2026a)	40.691	2.086	1.1035
VGGT (Wang et al., 2025a)	20.914	3.159	1.6396
ZipMap (Jin et al., 2026)	28.230	2.792	0.9454
LingBot-MAP (Chen et al., 2026a)	8.288	1.818	1.4152
VGGT-Long (Deng et al., 2025)	22.713	1.449	0.2537
VGGT-Align (Zhang et al., 2026b)	25.584	1.714	2.3608
VGGT-Omega (Wang et al., 2026b)	18.555	1.820	0.9878
π^3 (Wang et al., 2025c)	24.029	1.592	0.3614
GlueMap (Pan et al., 2026)	45.281	1.561	1.9132
Scal3R (Xie et al., 2026)	27.612	1.376	0.1703
HIP3R (LingBot-MAP)	**4.017**	1.055	0.1824
HIP3R (VGGT-Omega)	4.906	**0.590**	**0.0622**

4.2 Controlled Long-Horizon Reconstruction

We compare HIP3R with VGGT-Long under identical VGGT-Omega windows ($60/30$). Matched windows isolate the reconstruction algorithm from model capacity on KITTI, VBR, and Oxford. Table 2 reports ATE and the long-horizon mean, including sequential and loop-enabled VGGT-Long (Deng et al., 2025), VGGT-Align (Zhang et al., 2026b), and GlueMap (Pan et al., 2026).

Under matched VGGT-Omega windows, HIP3R achieves lower ATE than both VGGT-Long variants on all three splits. Loop closures reduce VGGT-Long ATE from 110.3 to 10.19 m on KITTI

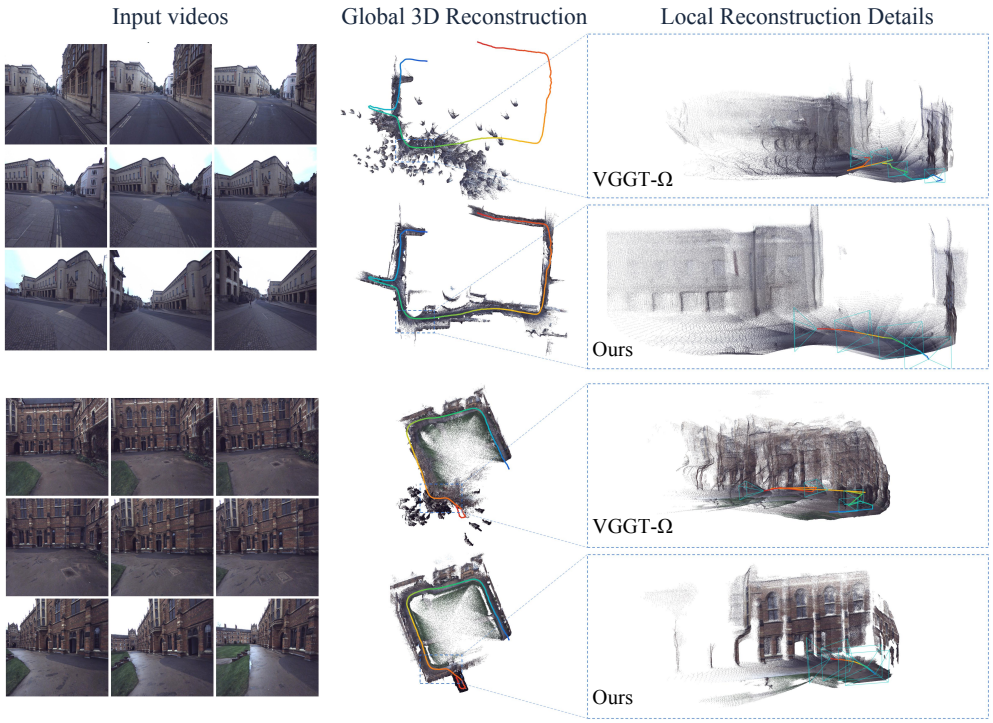


Figure 3: Qualitative comparison between VGGT-Omega chunk placement and HIP3R.

and from 85.50 to 41.27 m on VBR. HIP3R further reduces these errors to 9.730 and 22.44 m, nearly halving the remaining VBR error. VGGT-Long applies loops as rigid $\mathrm{Sim}(3)$ transforms to entire chunks. HIP3R instead jointly optimizes a sparse global layout graph with progressive loop scheduling. Appendix K shows that the complete solver further improves accuracy and convergence efficiency, while Appendix F reports stage-wise runtimes.

Table 2: Trajectory comparison on KITTI, VBR, and Oxford, alongside the long-horizon mean.

		ATE ↓			
Backend	**Method**	**Oxford**	**KITTI**	**VBR**	**Mean**
VGGT-Omega	Baseline	9.861	77.42	41.91	43.06
	VGGT-Long (w/o loops)	1.777	110.3	85.50	65.86
	VGGT-Long (w/ loops)	1.665	10.19	41.27	17.71
	+ HIP3R (Ours)	**1.579**	**9.730**	**22.44**	**11.25**
VGGT	VGGT-Long (released)	9.451	48.84	99.83	52.71
	VGGT-Align	8.567	69.80	99.87	59.41

4.3 End-to-End Pose Accuracy and 3D Geometry

Inconsistent poses place walls and corridors at conflicting locations, causing duplicated surfaces and ghosting. Table 3 reports pose AUC and surface metrics on both backends. Global layout alignment removes duplicate surfaces from unaligned chunks. Shared-camera trajectory refinement then yields one trajectory for those maps. With VGGT-Omega, HIP3R reduces Chamfer by 64%, from 0.0759 to 0.0272 m, and raises F1 from 65.92 to 91.26.

4.4 Qualitative Dense 3D Reconstruction

Figure 3 compares VGGT-Omega and HIP3R (VGGT-Omega) on Oxford Bodleian Library and Keble College. Sequential placement drifts across long temporal gaps, reconstructing unclosed loops as

offset surfaces several meters apart. Coarse layout optimization aligns distant revisits and collapses these surfaces onto single facades. As shown in the local views, shared-camera refinement further reconciles overlapping camera predictions. Figure 4 shows four long sequences with identical VGGT-Omega points. Native placement stretches streets into duplicated structures, while HIP3R recovers closed street networks with lower trajectory error.

4.5 ABLATION STUDIES

Ablations isolate point sampling, progressive loop scheduling, and placement-motion interaction.

Point sampling bound. Table 4 reports point sampling on cached predictions. Bounding sampled points cuts graph build and solve time from 12232.18 to 2004.20 s (83.62%), while ATE changes from 21.7751 to 21.7569 m. Total time includes measurement construction and layout solves.

Table 3: Pose AUC and dense geometry on both backends.

Method	**AUC@5** ↑	**AUC@15** ↑	**AUC@30** ↑	**Chamfer** ↓	**F1** ↑
VGGT-Omega	29.77	50.24	63.22	0.0759	65.92
HIP3R (VGGT-Omega)	50.75	67.98	75.21	0.0272	91.26
LingBot-MAP	17.83	38.61	53.19	0.1423	61.13
HIP3R (LingBot-MAP)	22.01	51.35	66.90	0.1042	74.60

Progressive loop scheduling and solver stabilization. The remaining rows of Table 4 add residual-fit weighting, progressive loop scheduling, and the complete solver. Progressive loop scheduling reduces graph build and solve time from 2077.49 to 1477.59 s. The complete configuration reaches 17.4708 m ATE in 753.43 s, with a 41.28 s final solve. The complete solver further combines Huber robust loss, adaptive continuation, and active-bound restarts. Huber loss suppresses outlier residuals, adaptive continuation avoids abruptly activating difficult long-span constraints around dense revisit clusters, and active-bound restarts recenter the log-scale bounds when they become active.

Table 4: Ablation of point sampling and progressive layout solver on extended sequences. Rows add components cumulatively. Appendix K details all test conditions.

Configuration	ATE ↓	Graph Build & Solve (s) ↓	Final solve (s)
Style control	21.7751	12232.18	1404.54
+ Bounded sampling	21.7569	2004.20	1345.43
+ Fit weighting	21.7770	2077.49	1387.14
+ Progressive loops	21.5241	1477.59	688.02
Complete	17.4708	753.43	41.28

Table 5: Trajectory refinement on identical predictions and one coarse layout. Layout-only exports placed chunks. Single, Unary, Motion, and Shared use different camera models in the fine stage.

	VGGT-Omega			LingBot-MAP		
	ATE	RPE_{rot}	RPE_{trans}	ATE	RPE_{rot}	RPE_{trans}
Layout-only	4.9143	0.6412	0.0817	4.0517	1.1837	0.2126
Single	4.9079	0.6238	0.0695	4.0256	1.1131	0.1930
Unary	4.9067	0.6010	0.0674	4.0192	1.0871	0.1828
Motion	11.6776	**0.5877**	**0.0619**	6.5007	**1.0464**	**0.1816**
Shared	**4.9061**	0.5899	0.0622	**4.0168**	1.0549	0.1824

Interaction of placement priors and motion constraints. Table 5 compares four trajectory-refinement variants, Single, Unary, Motion, and Shared, under identical predictions and coarse layouts, with Layout Only as a reference. Single uses one prediction per frame. Unary averages overlapping pose observations without relative-motion factors. Motion uses within-chunk motion factors without placement priors. Shared combines both factors as in Eq. (6). On both backends, Shared reduces relative errors over Single and Unary, with a small ATE gain. Motion yields slightly

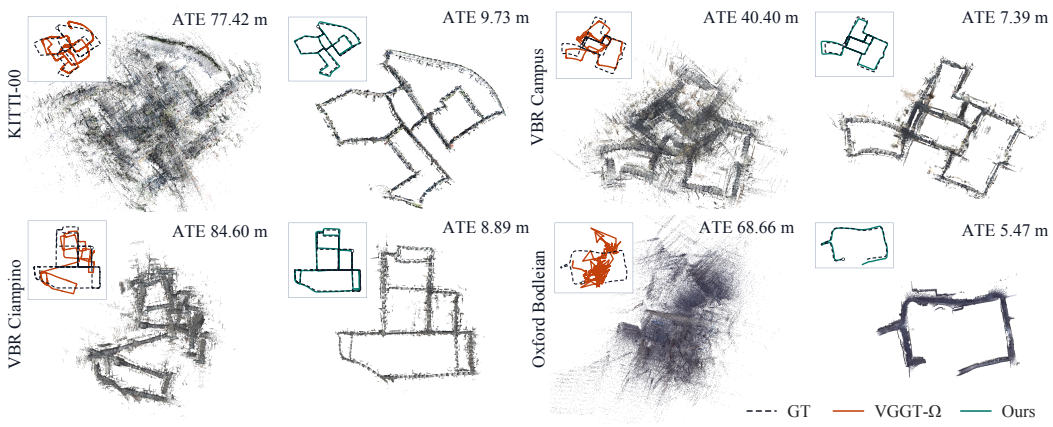


Figure 4: Trajectory comparison on four long sequences with identical VGGT-Omega points.

lower relative errors, but without placement priors, ATE increases to 11.678 and 6.501 m. Placement priors preserve the recovered layout, while motion factors improve local smoothness.

Table 6: Stage-wise results on identical VGGT-Omega predictions.

		Tracking Error			Runtime (s)	
Dataset	**Stage**	**ATE ↓**	**RPE$_{rot}$ (°) ↓**	**RPE$_{trans}$ ↓**	**Inference**	**Solver**
KITTI (T=4541)	Baseline	77.42	0.895	2.358	825.745	–
	+ Coarse	9.750$_{(-87\%)}$	0.202$_{(-77\%)}$	0.131$_{(-94\%)}$	896.785	5.40
	+ Fine	**9.730**$_{(-0.2\%)}$	**0.154**$_{(-24\%)}$	**0.093**$_{(-29\%)}$	896.785	0.08
VBR (T≈12,383)	Baseline	41.91	2.117	0.256	2283.438	–
	+ Coarse	22.44$_{(-46\%)}$	0.617$_{(-71\%)}$	0.075$_{(-71\%)}$	2511.460	200.0
	+ Fine	**22.44**$_{(-0.0\%)}$	**0.587**$_{(-5\%)}$	**0.059**$_{(-22\%)}$	2511.460	0.16
Oxford (T≈320)	Baseline	9.861	5.757	4.044	53.297	–
	+ Coarse	1.599$_{(-84\%)}$	0.579$_{(-90\%)}$	0.173$_{(-96\%)}$	54.919	0.03
	+ Fine	**1.579**$_{(-1\%)}$	**0.408**$_{(-30\%)}$	**0.104**$_{(-40\%)}$	54.919	0.02

Stage-wise results across layout recovery and trajectory refinement. Table 6 reports the Layout Only and Full outputs on identical VGGT-Omega predictions. Layout Only recovers global layout and scale, reducing ATE by 87% on KITTI, 84% on Oxford, and 46% on VBR. Full then adds shared-camera trajectory refinement on that fixed layout. It reduces RPE$_{rot}$ by 24% on KITTI, 30% on Oxford, and 5% on VBR, and RPE$_{trans}$ by 29%, 40%, and 22%.

Stage-wise results on dense 3D geometry. Layout alignment removes duplicate surfaces and provides most of the Chamfer gain. Shared-camera trajectory refinement improves local viewpoint consistency without changing predicted surface depth.

5 Conclusion

HiP3R turns overlapping local predictions into a globally consistent reconstruction of long video sequences through a hierarchical pose graph. The method first recovers global layout and scale through a chunk-level $\mathrm{Sim}(3)$ graph, then refines a shared-camera $\mathrm{SE}(3)$ graph with the chunk transforms held fixed. This separation allows distant loop closures to correct accumulated drift while reconciling conflicting predictions of the same camera across windows. Across seven benchmarks, HiP3R reduces VGGT-Omega trajectory error by 74%, from 18.555 to 4.906 m, and relative tracking drift by 94%, from 0.9878 to 0.0622 m. It also reduces surface reconstruction error by 64%, from 0.0759 to 0.0272 m, on the dense-geometry evaluation set. Extending this hierarchy to online updates and improving its robustness to unreliable loop closures are promising directions for future work.

AI USE STATEMENT

We used large language models to polish words and cross-check numerical tables. Every comparison and limitation was verified against tabulated evaluation results. We did not use generative tools to invent experiments, baselines, or citations. We take full responsibility for the text.

ETHICS STATEMENT

This work evaluates public camera-tracking benchmarks on outdoor and indoor scenes. It does not involve human subjects, biometric identification, or personal data collection.

REPRODUCIBILITY STATEMENT

Closed-form Jacobians and keyframe subgraphs appear in Appendices B and C. Protocol settings are detailed in Appendix D. Appendix E lists all datasets and splits. Every reported method row is evaluated locally on the seven-dataset protocol, including VBR. Code for both solver stages, window interfaces, and backend integrations will be released publicly alongside optimization hyperparameters and alignment settings.

REFERENCES

Dejan Azinović, Ricardo Martin-Brualla, Dan B. Goldman, Matthias Nießner, and Justus Thies. Neural RGB-D surface reconstruction. In *CVPR*, 2022.

Leonardo Brizi, Emanuele Giacomini, Luca Di Giammarino, Simone Ferrari, Omar Salem, Lorenzo De Rebotti, and Giorgio Grisetti. VBR: A vision benchmark in Rome. In *ICRA*, 2024.

Lin-Zhuo Chen, Jian Gao, Yihang Chen, Ka Leong Cheng, Yipengjing Sun, Liangxiao Hu, Nan Xue, Xing Zhu, Yujun Shen, Yao Yao, and Yinghao Xu. Geometric context transformer for streaming 3D reconstruction. *arXiv preprint arXiv:2604.14141*, 2026a.

Xingyu Chen, Yue Chen, Yuliang Xiu, Andreas Geiger, and Anpei Chen. TTT3R: 3D reconstruction as test-time training. In *ICLR*, 2026b.

A. R. Curtis, M. J. D. Powell, and J. K. Reid. On the estimation of sparse Jacobian matrices. *IMA Journal of Applied Mathematics*, 13(1):117–119, 1974.

Kai Deng, Zexin Ti, Jiawei Xu, Jian Yang, and Jin Xie. VGGT-Long: Chunk it, loop it, align it – pushing VGGT's limits on kilometer-scale long RGB sequences. *arXiv preprint arXiv:2507.16443*, 2025.

Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? The KITTI vision benchmark suite. In *CVPR*, 2012.

Sergio Izquierdo and Javier Civera. Optimal transport aggregation for visual place recognition. In *CVPR*, 2024.

Haian Jin, Rundi Wu, Tianyuan Zhang, Ruiqi Gao, Jonathan T. Barron, Noah Snavely, and Aleksander Holynski. ZipMap: Linear-time stateful 3D reconstruction via test-time training. In *CVPR*, 2026.

Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics*, 36(4), 2017.

Yushi Lan, Yihang Luo, Fangzhou Hong, Shangchen Zhou, Honghua Chen, Zhaoyang Lyu, Shuai Yang, Bo Dai, Chen Change Loy, and Xingang Pan. STream3R: Scalable sequential 3d reconstruction with causal transformer. In *ICLR*, 2026.

Vincent Leroy, Yohann Cabon, and Jerome Revaud. Grounding image matching in 3d with MASt3R. In *ECCV*, 2024.

Zizun Li, Jianjun Zhou, Yifan Wang, Haoyu Guo, Wenzheng Chang, Yang Zhou, Haoyi Zhu, Junyi Chen, Chunhua Shen, and Tong He. WinT3R: Window-based streaming reconstruction with camera token pool. In *ICLR*, 2026.

Xuanyi Liu, Chunan Yu, Deyi Ji, Qi Zhu, Lingyun Sun, Xuanfu Li, Jin Ma, Tianrun Chen, and Lanyun Zhu. StreamCacheVGGT: Streaming visual geometry transformers with robust scoring and hybrid cache compression. *arXiv preprint arXiv:2604.15237*, 2026.

Linfei Pan, Johannes Schönberger, and Marc Pollefeys. Global structure-from-motion meets feed-forward reconstruction. *arXiv preprint arXiv:2605.26103*, 2026.

Thomas Schöps, Johannes L. Schönberger, Silvano Galliani, Torsten Sattler, Konrad Schindler, Marc Pollefeys, and Andreas Geiger. A multi-view stereo benchmark with high-resolution images and multi-camera videos. In *CVPR*, 2017.

Jamie Shotton, Ben Glocker, Christopher Zach, Shahram Izadi, Antonio Criminisi, and Andrew Fitzgibbon. Scene coordinate regression forests for camera relocalization in RGB-D images. In *CVPR*, 2013.

Joan Solà, Jeremie Deray, and Dinesh Atchuthan. A micro Lie theory for state estimation in robotics. *arXiv preprint arXiv:1812.01537*, 2018.

Yifu Tao, Miguel Ángel Muñoz-Bañón, Lintong Zhang, Jiahao Wang, Lanke Frank Tarimo Fu, and Maurice Fallon. The Oxford Spires dataset: Benchmarking large-scale LiDAR-visual localisation, reconstruction and radiance field methods. *The International Journal of Robotics Research*, 2025.

Shinji Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(4):376–380, 1991.

Chung-Shien Brian Wang, Christian Schmidt, Jens Piekenbrinck, and Bastian Leibe. Block-sparse global attention for efficient multi-view geometry transformers. In *CVPR*, 2026a.

Hengyi Wang and Lourdes Agapito. 3D reconstruction with spatial memory. In *3DV*, 2025.

Hengyi Wang and Lourdes Agapito. AMB3R: Accurate feed-forward metric-scale 3D reconstruction with backend. In *CVPR*, 2026.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. VGGT: Visual geometry grounded transformer. In *CVPR*, 2025a.

Jianyuan Wang, Minghao Chen, Shangzhan Zhang, Nikita Karaev, Johannes Schönberger, Patrick Labatut, Piotr Bojanowski, David Novotny, Andrea Vedaldi, and Christian Rupprecht. VGGT-Ω. In *CVPR*, 2026b.

Qianqian Wang, Yifei Zhang, Aleksander Holynski, Alexei A. Efros, and Angjoo Kanazawa. Continuous 3d perception model with persistent state. In *CVPR*, 2025b.

Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. DUSt3R: Geometric 3d vision made easy. In *CVPR*, 2024.

Yifan Wang, Jianjun Zhou, Haoyi Zhu, Wenzheng Chang, Yang Zhou, Zizun Li, Junyi Chen, Jiangmiao Pang, Chunhua Shen, and Tong He. π^3: Permutation-equivariant visual geometry learning. *arXiv preprint arXiv:2507.13347*, 2025c.

Tao Xie, Peishan Yang, Yudong Jin, Yingfeng Cai, Wei Yin, Weiqiang Ren, Qian Zhang, Wei Hua, Sida Peng, Xiaoyang Guo, and Xiaowei Zhou. Scal3R: Scalable test-time training for large-scale 3D reconstruction. In *CVPR*, 2026.

Zhuang Xiong, Chen Zhang, Qingshan Xu, and Wenbing Tao. VGGT-Motion: Motion-aware calibration-free monocular slam for long-range consistency. *arXiv preprint arXiv:2602.05508*, 2026.

Junyi Zhang, Charles Herrmann, Junhwa Hur, Chen Sun, Ming-Hsuan Yang, Forrester Cole, Trevor Darrell, and Deqing Sun. LoGeR: Long-context geometric reconstruction with hybrid memory. *arXiv preprint arXiv:2603.03269*, 2026a.

Wei Zhang, Yihang Wu, Songhua Li, and Qi Wang. VGGT-Align: Bridging local reconstruction and global consistency for long-sequence 3d reconstruction. *arXiv preprint arXiv:2608.15260*, 2026b.

Dong Zhuo, Wenzhao Zheng, Jiahe Guo, Yuqi Wu, Jie Zhou, and Jiwen Lu. Streaming visual geometry transformer. In *ICLR*, 2026.

Zichen Zou, Xiaosong Jia, Zuxuan Wu, and Yu-Gang Jiang. Attention Itself Could Retrieve. RetrieveVGGT: Training-free long context streaming 3d reconstruction via query-key similarity retrieval. *arXiv preprint arXiv:2605.09644*, 2026.

A APPENDIX

In this supplementary material, we provide mathematical derivations, implementation protocols, and extended experimental analyses. Appendix B details closed-form Jacobian derivations and graph coloring. Appendix C specifies the keyframe subgraph and residual interpolation for long sequences. Appendix D details public solver settings and optimization hyperparameters. Appendix E provides dataset compositions and evaluation splits across all seven benchmarks. Appendix F breaks down stage-wise execution times across competing methods. Appendices G and H report per-dataset tracking and dense 3D geometry results. Finally, Appendices I through M present window diagnostics, solver ablations, and dense geometry comparisons.

B GRAPH COLORING AND CLOSED-FORM JACOBIAN BLOCKS

This appendix provides mathematical derivations for the sparse Jacobian blocks in Section 3.3. Derivative identities follow standard Lie-group calculus (Solà et al., 2018). We first analyze Jacobian sparsity and compare finite-difference graph coloring with analytic derivatives. We then derive exact closed-form blocks for both $\mathrm{SE}(3)$ and $\mathrm{Sim}(3)$ residuals.

Jacobian sparsity and manifold structure. Nonlinear least-squares iterations linearize $r(x)$ at current poses. Jacobian $J = \partial r / \partial x$ has one row per residual coordinate and one column per pose parameter. For a 20,000-frame graph, n reaches 10^5 and m spans millions. A dense J consists mostly of zeros. A relative residual between frames a and b depends only on those two cameras under fixed chunk transforms.

Graph coloring baseline for finite differences. Without closed-form derivatives, J can be approximated via finite differences. Evaluating columns individually costs n residual passes. Sparsity enables a more efficient approach. Two columns are structurally orthogonal if they share no residual rows. Perturbing orthogonal variables simultaneously isolates each column after differencing. Each residual experiences at most one perturbation (Curtis et al., 1974). Coloring the column-intersection graph groups mutually orthogonal columns together. The number of residual sweeps equals the chromatic number. This depends on graph topology instead of total parameter count. On a 735-frame scene, coloring cuts post-processing from 10,228 to 172 s. The closed-form derivatives below eliminate these colored sweeps for $\mathrm{SE}(3)$ residuals.

Lie group calculus and rotation identities. Let $[u]_\times$ denote the skew-symmetric matrix where $[u]_\times v = u \times v$. Let Exp and Log denote the $\mathrm{SO}(3)$ exponential and logarithm. We use three standard Lie identities (Solà et al., 2018).

$$\mathrm{Exp}(\phi + \delta) \approx \mathrm{Exp}(\phi)\,\mathrm{Exp}\big(J_r(\phi)\,\delta\big), \tag{7}$$

$$\begin{aligned} \mathrm{Log}\big(R\,\mathrm{Exp}(\delta)\big) &\approx \mathrm{Log}(R) + J_r^{-1}\big(\mathrm{Log}(R)\big)\,\delta, \\ \mathrm{Log}\big(\mathrm{Exp}(\delta)\,R\big) &\approx \mathrm{Log}(R) + J_l^{-1}\big(\mathrm{Log}(R)\big)\,\delta, \end{aligned} \tag{8}$$

$$A\,\mathrm{Exp}(u) = \mathrm{Exp}(Au)\,A, \qquad A \in \mathrm{SO}(3). \tag{9}$$

The right Jacobian and its inverse have the closed forms ($\theta = \|\phi\|$)

$$J_r(\phi) = I - \tfrac{1-\cos\theta}{\theta^2}[\phi]_\times + \tfrac{\theta-\sin\theta}{\theta^3}[\phi]_\times^2, \qquad J_r^{-1}(\phi) = I + \tfrac{1}{2}[\phi]_\times + \Big(\tfrac{1}{\theta^2} - \tfrac{1+\cos\theta}{2\theta\sin\theta}\Big)[\phi]_\times^2, \tag{10}$$

and $J_l^{-1}(\omega) = J_r^{-1}(\omega)^\top$. Identity (7) differentiates global rotation vectors, where frame t carries $x_t = (\phi_t, \tau_t) \in \mathbb{R}^6$ with $R_t = \mathrm{Exp}(\phi_t)$.

Closed-form factorized $\mathrm{SE}(3)$ residual. Consider a fine edge between source a and target b with measurement $M = (R_M, t_M)$. Expanding $M^{-1}F_b^{-1}F_a$ into rotation and translation yields

$$r_R = \mathrm{Log}\big(\underbrace{R_M^\top R_b^\top R_a}_{E_R}\big), \qquad r_t = R_M^\top\big(R_b^\top(\tau_a - \tau_b) - t_M\big). \tag{11}$$

This factorization provides favorable sparsity. Specifically, r_R never depends on translation, while r_t never depends on source rotation. A dense 6×12 local Jacobian exceeds residual requirements. After the coarse stage, M remains constant.

Analytic rotation derivatives. Perturbing $\phi_a \rightarrow \phi_a + \delta$ and applying (7) gives $E_R \rightarrow E_R \operatorname{Exp}(J_r(\phi_a)\delta)$, so by (8)

$$\frac{\partial r_R}{\partial \phi_a} = J_r^{-1}(r_R)\, J_r(\phi_a). \tag{12}$$

Perturbing ϕ_b gives $R_b^\top \rightarrow \operatorname{Exp}(-J_r(\phi_b)\delta)R_b^\top$. Transporting the perturbation through $R_M^\top$ with (9) yields $E_R \rightarrow \operatorname{Exp}(-R_M^\top J_r(\phi_b)\delta)\, E_R$ and hence

$$\frac{\partial r_R}{\partial \phi_b} = -\, J_l^{-1}(r_R)\, R_M^\top J_r(\phi_b). \tag{13}$$

Each block multiplies at most three 3×3 matrices. These include the inverse log Jacobian, a frame-transport rotation, and the target right Jacobian.

Analytic translation derivatives. The translation residual is linear in translations.

$$\frac{\partial r_t}{\partial \tau_a} = R_M^\top R_b^\top, \qquad \frac{\partial r_t}{\partial \tau_b} = -R_M^\top R_b^\top, \tag{14}$$

depending on rotation only through the target. Using $-[u]_\times v = [v]_\times u$,

$$\frac{\partial r_t}{\partial \phi_b} = R_M^\top \left[R_b^\top(\tau_a - \tau_b)\right]_\times J_r(\phi_b), \qquad \frac{\partial r_t}{\partial \phi_a} = 0. \tag{15}$$

A fine edge contributes five nonzero 3×3 blocks. The prior residual from Eq. (5) with placement $Q = (R_Q, t_Q)$ factorizes similarly. Specifically, $r_R^p = \operatorname{Log}(R_Q^\top R_g)$ and $r_t^p = R_Q^\top(\tau_g - t_Q)$. This yields two diagonal blocks.

$$\frac{\partial r_R^p}{\partial \phi_g} = J_r^{-1}(r_R^p)\, J_r(\phi_g), \qquad \frac{\partial r_t^p}{\partial \tau_g} = R_Q^\top. \tag{16}$$

Sparsity pattern assembly and reuse. Quantities R_t and $J_r(\phi_t)$ appear in every factor incident to frame t. They are computed once per iteration in batch and shared. Per-edge terms $J_r^{-1}(r_R)$ and $J_l^{-1}(r_R)$ are evaluated in batch from current residuals. The CSR sparsity structure depends only on graph topology, requiring a single initialization. Subsequent iterations update only numeric block values. Jacobian construction costs few batched 3×3 multiplications per edge, scaling linearly without residual sweeps.

Coarse $\mathrm{Sim}(3)$ residual formulation. Let $A_i = (s_i, R_i, t_i)$ and $A_j = (s_j, R_j, t_j)$. Relative similarity $\widehat{Z}_{ij} = (\hat{s}, \hat{R}, \hat{t})$ comes from confidence-weighted Umeyama alignment on overlap or loop pointmaps. The coarse residual in Eq. (2) is

$$E_{ij} = \widehat{Z}_{ij}^{-1} A_j^{-1} A_i, \qquad \tilde{r}_{ij}^c = \sqrt{w_{ij}}\, r_{ij}^c = \sqrt{w_{ij}} \begin{bmatrix} \operatorname{Log}_{\mathrm{SO}(3)} R(E_{ij}) \\ \tau(E_{ij}) \\ \log s(E_{ij}) \end{bmatrix} \in \mathbb{R}^7, \tag{17}$$

ordered as $(\omega,\, \tau,\, \log s)$ with $\omega \in \mathbb{R}^3$ and translation $\tau \in \mathbb{R}^3$. Edge weights apply prior to the Huber penalty. Fixing A_1 leaves $7(K-1)$ active parameters. Huber loss is coordinate-wise with $\delta = 1$. Adaptive Jacobi scaling normalizes parameter updates across rotation, translation, and scale.

Coarse $\mathrm{Sim}(3)$ blocks via automatic differentiation. Similarity composition couples scales, rotations, and translations between endpoints. We differentiate per-edge residuals using vectorized float64 automatic differentiation. This outputs one 7×14 block per edge, assembled into CSR format. Both closed-form $\mathrm{SE}(3)$ and $\mathrm{Sim}(3)$ blocks match numerical derivatives within 1.1×10^{-8}. Because the fine stage holds $A^\star$ fixed, chunk columns drop before the $\mathrm{SE}(3)$ solve.

C Long-sequence keyframe subgraph

For sequences exceeding 2000 frames, optimizing camera variables at every single frame is computationally unnecessary. Camera motion across adjacent video frames is temporally smooth. Therefore, the fine stage solves the objective over an active keyframe subgraph. Dense per-frame trajectory smoothness is recovered via Lie-algebraic residual interpolation. This appendix specifies selection rules, interpolation mechanics, and stride evaluations.

Keyframe selection rules. Let T denote frame count and $s{=}5$ the default stride. Target indices include $\{0, s, 2s, \ldots\}$ and $\{T{-}1\}$. Each chunk $[a, b)$ also contributes boundary frames a and $b{-}1$. For adjacent chunk overlap $\mathcal{O} = \{t : t \in C_k \cap C_{k+1}\}$, we check existing samples. If fewer than $m = \min(3, |\mathcal{O}|)$ indices lie in $\mathcal{O}$, we insert m linearly spaced frames. The fine graph builds exclusively over this selected index set. Specifically, reduced chunks retain only keyframe cameras. Within-chunk edges connect consecutive keyframes, and placement priors attach only to retained nodes. Omitted frames introduce no variables or residuals.

Lie-algebraic residual interpolation. Let $Q_t = Q_{k,i}(A^\star)$ denote the coarse pose for frame t. A solved keyframe pose F_{t_i} defines correction $\Delta_{t_i} = F_{t_i} Q_{t_i}^{-1}$ in world coordinates. Between keyframes $t_i < t_{i+1}$ with weight $\alpha = (t - t_i)/(t_{i+1} - t_i)$,

$$R(\Delta_t) = \mathrm{Slerp}\big(R(\Delta_{t_i}), R(\Delta_{t_{i+1}}), \alpha\big), \qquad \tau(\Delta_t) = (1-\alpha)\,\tau(\Delta_{t_i}) + \alpha\,\tau(\Delta_{t_{i+1}}), \tag{18}$$

and $F_t = \Delta_t\, Q_t$. Rather than interpolating raw camera poses in local space, our method directly interpolates residual corrections in world coordinates.

Stride sensitivity and comparisons. Overlap averaging computes consensus poses before applying residual interpolation, omitting fine optimization. Uniform stride $s{=}5$ is our default setting. Table 7 compares uniform $s{=}10$ with motion-spaced $s{=}10$. Motion spacing allocates targets by accumulated motion instead of frame indices. Table 8 extends this evaluation across three long sequences. Table 9 evaluates dense versus keyframe graphs on KITTI.

Table 7: Keyframe subgraph on KITTI seq00 after the same coarse $A^\star$. Subscripts are versus overlap averaging (no fine least-squares). Uniform stride 5 is the default.

Refinement graph	ATE ↓	RPE_{rot} ↓	$\text{RPE}_{\text{trans}}$ ↓	$\lvert\mathcal{K}\rvert$	t_{f} (s)
Overlap averaging	9.735	0.202	0.131	—	—
Uniform stride 10	9.728	$0.158_{(-22\%)}$	$0.094_{(-28\%)}$	605	0.055
Motion stride 10	9.729	$0.156_{(-23\%)}$	$0.094_{(-28\%)}$	725	0.104
Uniform stride 5	9.730	$0.154_{(-24\%)}$	$0.093_{(-29\%)}$	1059	0.075

Table 8: The same keyframe comparison on additional long sequences.

		Consensus		Uniform $s{=}10$		Uniform $s{=}5$	
Scene	T	RPE_{rot}	$\text{RPE}_{\text{trans}}$	RPE_{rot}	$\text{RPE}_{\text{trans}}$	RPE_{rot}	$\text{RPE}_{\text{trans}}$
campus_train0	12,042	0.204	0.129	0.122	0.118	0.119	0.117
spagna_train0	14,141	0.542	0.048	0.535	0.035	0.535	0.034
downtown2	2200	3.509	0.083	3.507	0.082	3.507	0.082

Uniform stride $s{=}5$ remains the default setting. It achieves lower relative errors than $s{=}10$. On KITTI, solve times are 0.075 and 0.055 s, respectively.

Table 9: Dense fine graph versus uniform stride 5 on identical VGGT-Omega (Wang et al., 2026b) predictions and coarse $A^\star$. Courthouse and downtown1 have lengths between 1000 and 2000 frames. KITTI represents the long-sequence setting.

		Dense F_t			Uniform $s{=}5$		
Scene	T	ATE	RPE_{rot}	$\text{RPE}_{\text{trans}}$	ATE	RPE_{rot}	$\text{RPE}_{\text{trans}}$
Courthouse	1106	2.599	10.82	0.517	2.600	10.81	0.517
downtown1	1427	3.223	3.940	0.0875	3.223	3.940	0.0876
KITTI seq00	4541	9.731	0.144	0.086	9.730	0.154	0.093

Table 9 compares dense and keyframe graphs. On Courthouse and downtown1 (1106 and 1427 frames), stride 5 yields errors comparable to the dense graph. On KITTI (4541 frames), dense solving improves RPE_{rot} and $\text{RPE}_{\text{trans}}$ by 6% and 8% with unchanged ATE. The default configuration retains the keyframe subgraph on long sequences.

D HYPERPARAMETERS AND PROTOCOL

This appendix documents all solver configurations, termination tolerances, and optimization schedules. Table 10 provides public solver settings and protocol details across backends. The main benchmark uses VGGT-Omega windows of $60/30$ and LingBot-MAP windows of $120/32$. Table 18 examines variations in these window parameters.

Table 10: Public solver hyperparameters and optimization configurations across backends. We use the public configurations of VGGT-Omega (Wang et al., 2026b), LingBot-MAP (Chen et al., 2026a), and SALAD (Izquierdo & Civera, 2024).

Quantity	**Value**
VGGT-Omega default window	C=60, O=30 frames
LingBot-MAP default window	C=120, O=32, windowed inference, decoded C2W, depth unprojection
Loop window	20 frames per endpoint
SALAD retrieval	similarity 0.85, top-5, NMS 25 frames, one pair per chunk pair (first hit)
Loop-constraint addition	4 configured span-ordered stages, expanded to at most 8 by adaptive continuation
Keyframe subgraph ($T > 2000$)	uniform stride 5, all chunk endpoints, ≥ 3 linspace overlap frames if the uniform set is short, rotation and translation interpolation of Δ_t (Appendix C)
Refinement weights (Eq. (6))	λ_f=1, λ_p=0.25, fixed across backends and windows
Coarse $\mathrm{Sim}(3)$	Huber f_{scale}=1, adaptive Jacobi scaling (`x_scale='jac'`), `max_nfev`=5000, `ftol`=`xtol`=`gtol`=10^{-6}
Coarse continuation	intermediate stages (100 evals), adaptive full solve (500), full fallback (5000)
Active-bound restarts	at most 2 passes of 1000 evaluations, with log-scale bounds $\pm \log 100$ around initialization
Shared-camera $\mathrm{SE}(3)$	$A^\star$ held after coarse, `max_nfev`=100, `ftol`=10^{-8}, `translation_scale`=1, λ_f=1, λ_p=0.25 are fixed.
Jacobian	closed-form 3×3 $\mathrm{SE}(3)$ blocks in CSR, coarse $\mathrm{Sim}(3)$ by float64 automatic differentiation

Overlap O is an absolute frame count, not $C/2$. Adjacent alignment subsamples to 2×10^5 points. Coarse graph optimization uses an evaluation limit of up to 5000 iterations with adaptive Jacobi scaling. This normalizes parameter sensitivities across rotation, translation, and scale dimensions. The fine camera pose graph converges within 100 evaluations. Optimization terminates when relative parameter changes fall below tolerance (`ftol` = `xtol` = 10^{-6}). Inference and alignment overlaps are set to 30 frames for VGGT-Omega and 32 frames for LingBot-MAP. Subsampling correspondences to 2×10^5 points bounds alignment memory while preserving sufficient geometric constraints for reliable convergence.

Adaptive schedule and restarts. Let L denote the number of loop edges, $n_{\min}$ the shortest span count, and $S = 4$ the configured stage count. When loop spans vary and $n_{\min} S \geq L(S-1)$, the schedule expands to $\min(2S, L)$ stages. Longer-span constraints then enter through additional intermediate layouts. Intermediate stages use 100 evaluations, while the adaptive full-graph solve uses 500. If the full solve reaches its evaluation ceiling without converging, the solver invokes a fallback with extended iterations. Active scale bounds trigger the restart routine instead. Restarts warm-start from the latest estimate and recenter scale bounds. They continue while the problem remains unconverged and bounds remain active, capped at two passes of 1000 evaluations.

E DATASETS AND SPLITS

We evaluate HIP3R across seven public benchmarks spanning 61 diverse sequences. Table 11 lists dataset compositions, sensor domains, and evaluation properties.

Indoor room-scale and multi-scale sequences. These three datasets evaluate performance across local and multi-scale settings with varying sequence lengths and scene complexity. Under the default VGGT-Omega $60/30$ window, 10 of the 11 ETH3D still sequences fit within a single chunk, so the layout and shared-camera graphs remain inactive, and the Native, Layout Only, and Full

Table 11: Dataset composition and experimental splits.

Dataset	Domain	#Seq.	What it tests
7-Scenes	Indoor RGB-D		Textureless rooms, motion blur
ETH3D	Indoor and outdoor	38	Mixed-scale laser-scanned scenes
Neural RGB-D	Indoor RGB-D		Cluttered rooms, fine geometry
Tanks and Temples	Outdoor structures	6	Large objects and buildings
Oxford Spires	Campus indoor and outdoor	9	Loops and scale change ($T\approx320$)
KITTI seq00	Driving	1	Long outdoor loop ($T=4541$)
VBR	City-scale Rome	7	Extreme length (T up to 18,846)
Total		61	

trajectories coincide. Neural RGB-D spans 2–11 chunks per sequence, while 7-Scenes spans 3–6. **7-Scenes** (Shotton et al., 2013) consists of seven Kinect-captured rooms: Chess, Fire, Heads, Office, Pumpkin, Kitchen, and Stairs. **ETH3D** (Schöps et al., 2017) includes high-resolution indoor and outdoor still sequences with laser-scanner ground truth, covering offices, reliefs, facades, playgrounds, and gardens. **Neural RGB-D** (Azinović et al., 2022) contains nine room-scale indoor scenes with structured-light depth, including kitchens, staircases, and apartments.

Outdoor structures and buildings. Tanks and Temples (Knapitsch et al., 2017) evaluates Barn, Caterpillar, Church, Ignatius, Meeting Room, and Truck.

Extended traversals and driving loops. Oxford Spires (Tao et al., 2025) provides nine campus sequences with LiDAR-inertial ground truth, subsampled to $T\approx320$ frames. The sequences mix outdoor courtyards, interiors, and staircases, with revisits after long temporal gaps. **KITTI** (Geiger et al., 2012) uses odometry sequence 00 only ($T=4541$), a standard benchmark for long driving loops. **VBR** (Brizi et al., 2024) contains seven long urban sequences through Rome, $8{,}815$–$18{,}846$ frames per sequence (dataset-mean $T=12{,}383$).

Aggregation. To ensure balanced representation across datasets of varying sequence lengths, overall benchmark metrics are macro-averaged across all seven datasets. Global trajectory error (ATE) reflects large gains on kilometer-scale driving and city sequences (KITTI and VBR). In these extended environments, sequential tracking drift is most pronounced. Relative pose metrics (RPE_{rot} and $\text{RPE}_{\text{trans}}$) measure local frame-to-frame consistency across all environments.

F Reported pipeline-stage timings

In this appendix, we analyze computational latency across individual stages of the reconstruction pipeline. Table 12 reports execution times across individual stages of the reconstruction pipeline on a single GPU. For HiP3R, this includes baseline chunk inference (t_{g}^{N}), loop-window inference ($t_{\text{loop}} = t_{\text{g}} - t_{\text{g}}^{\text{N}}$), and CPU graph work ($t_{\text{graph}}$). The t_{graph} column includes bounded point sampling, weighted Umeyama/IRLS edge construction, two-endpoint loop registration, and both the coarse layout and shared-camera refinement solves. It is not shared-camera refinement solver time alone. Table 4 further splits graph build & solve from the final least-squares call. These component sums isolate core algorithmic execution times, excluding asynchronous disk I/O, retrieval, and file export. For VGGT-Long, reported components include chunk inference, loop-window inference when enabled, and pairwise $\text{Sim}(3)$ alignment. Pairwise alignment and global graph optimization involve distinct operations. Therefore, we report individual stage latencies rather than an unstructured aggregate runtime. Controlled efficiency evidence appears in Table 4. We evaluate both default and sequential VGGT-Long pipelines on identical VGGT-Omega windows. We also report the released VGGT-Long system. GlueMap component times include Pi3 star forwards, tracking, and coarse mapping on one GPU.

The large alignment times in VGGT-Long show pairwise 3D point alignment to be a dominant computational bottleneck. By contrast, HiP3R accelerates pairwise alignment via bounded sampling while using a fast sparse graph solver. Dense point tracking accounts for most of GlueMap runtime. This represents a distinct computational strategy from windowed feed-forward reconstruction.

Table 12: Stage-wise execution times in seconds across evaluated methods. Component sums represent per-stage latency rather than unstructured end-to-end runtimes. Oxford and VBR entries are dataset aggregates.

Method	Component	Split		
		Oxford (T≈320)	**KITTI seq00** (T=4541)	**VBR** (T≈12,383)
HIP3R (Ours)	Local baseline chunk inference ($t_{\mathrm{g}}^{\mathrm{N}}$)	53.3	825.7	2283.4
	Loop-window inference (t_{loop})	1.6	71.0	228.0
	CPU pose graph solve (t_{graph})	0.05	5.5	200.0
	Component sum	55.0	902.3	2711.5
VGGT-Long *(w/ loops), on Omega windows*	Chunk inference	57.2	924.2	2664.0
	Loop-window inference	0.5	75.3	252.7
	Chunk alignment ($\mathrm{Sim}(3)$ IRLS)	139.9	2882.0	9489.5
	Component sum	197.6	3881.5	12,406.2
VGGT-Long *(w/o loops), on Omega windows*	Chunk inference	67.5	910.1	2919.0
	Chunk alignment ($\mathrm{Sim}(3)$ IRLS)	185.6	2457.1	8237.7
	Component sum	253.1	3367.2	11,156.7
VGGT-Long *(released)*	Chunk inference	67.4	794.6	2329.4
	Chunk alignment ($\mathrm{Sim}(3)$ IRLS)	169.2	2228.8	5036.7
	Component sum	236.6	3023.3	7366.2
GlueMap *(Pi3 coarse)*	Star forward (Pi3)	280.9	2913.2	7803.2
	VGGSfM tracking	996.2	11,947.8	38,762.2
	Coarse mapping	215.2	1935.8	5077.2
	Component sum	1492.3	16,796.8	51,642.6

G PER-DATASET COMPARISON

Table 1 in the main text summarizes macro-averaged tracking metrics across all seven public benchmarks. Tables 13, 14, and 15 report detailed per-dataset tracking errors for ATE, $\mathrm{RPE}_{\mathrm{rot}}$, and $\mathrm{RPE}_{\mathrm{trans}}$, respectively. In each column, bold and underline mark the best and second-best results across all evaluated methods. On VGGT-Omega, HIP3R improves relative rotation and translation accuracy across all seven datasets. Trajectory ATE improvements are largest on extended outdoor traversals (KITTI and VBR). In these environments, global loop constraints effectively eliminate accumulated drift. On compact indoor scenes, baseline predictions already achieve centimeter-level trajectory accuracy. For LingBot-MAP, HIP3R acts as a modular post-processing method. It reduces macro-averaged ATE from 8.288 to 4.017 m.

H PER-DATASET DENSE 3D GEOMETRY

This appendix provides per-dataset surface reconstruction metrics for the dense geometry benchmark. The dense-geometry evaluation set comprises 38 sequences with laser ground truth. It includes 11 sequences from ETH3D, 9 from Neural RGB-D, and 18 from 7-Scenes. To weight each physical evaluation scene equally, Table 3 reports sequence-macro averages over these 38 sequences. The aggregate values in Table 3 reproduce from the per-dataset metrics in Tables 16 and 17 via

$$M_{\text{aggregate}} = \frac{11 \cdot M_{\text{ETH3D}} + 9 \cdot M_{\text{NRGBD}} + 18 \cdot M_{\text{7-Scenes}}}{38}. \tag{19}$$

Under unweighted dataset-macro averaging across the three datasets, VGGT-Omega baseline versus HIP3R scores are 0.0796 versus 0.0269 for Chamfer distance, and 65.97 versus 92.06 for F1 score. LingBot-MAP baseline versus HIP3R scores are 0.1439 versus 0.1014 for Chamfer distance, and 61.34 versus 74.58 for F1 score. Both aggregation protocols show substantial geometric gains. Table 21 isolates the relative geometric contributions of layout alignment and camera refinement.

Table 13: Per-dataset ATE on the seven public datasets (lower is better). NRGBD is Neural RGB-D, 7S is 7-Scenes, T&T is Tanks and Temples. Bold and underline mark the full-table best and second-best values in each column.

Method	7S	ETH	KITTI	NRGBD	Oxford	T&T	VBR
VGGT-Omega (Wang et al., 2026b)	**0.0370**	0.118	77.42	**0.0190**	9.861	0.518	41.914
π^3 (Wang et al., 2025c)	0.0574	<u>0.104</u>	68.22	0.0244	12.06	**0.467**	87.269
GlueMap (Pan et al., 2026)	0.0624	0.810	193.5	0.0232	16.65	0.521	105.4
Scal3R (Xie et al., 2026)	0.0720	0.236	86.25	0.0471	4.573	0.572	101.535
VGGT-Long (Deng et al., 2025)	0.0743	0.204	48.84	0.0374	9.451	0.550	99.832
VGGT-Align (Zhang et al., 2026b)	0.0731	0.209	69.80	0.0374	8.567	0.537	99.867
LingBot-MAP (Chen et al., 2026a)	0.0789	0.441	23.40	0.0554	2.391	0.518	31.129
ZipMap (Jin et al., 2026)	0.0702	0.236	127.22	0.0735	28.33	0.692	40.987
VGGT (Wang et al., 2025a)	0.0705	0.237	79.87	0.164	21.40	0.536	44.123
LoGeR (Zhang et al., 2026a)	0.118	0.370	174.96	0.138	5.522	1.078	102.650
TTT3R (Chen et al., 2026b)	0.104	1.189	46.93	0.191	12.51	1.302	104.199
CUT3R (Wang et al., 2025b)	0.296	1.376	190.19	0.546	16.26	1.839	103.748
3.107	106.120						
H1P3R (VGGT-Omega)	<u>0.0381</u>	**0.0546**	<u>9.730</u>	<u>0.0219</u>	**1.579**	<u>0.476</u>	<u>22.443</u>
H1P3R (LingBot-MAP)	0.0901	0.488	**6.063**	0.126	<u>2.346</u>	1.026	**17.978**

Table 14: Per-dataset RPE_{rot} on the same seven datasets (lower is better). Bold and underline mark the full-table best and second-best values in each column.

Method	7S	ETH	KITTI	NRGBD	Oxford	T&T	VBR
VGGT-Omega (Wang et al., 2026b)	<u>0.501</u>	0.868	0.895	0.146	5.757	2.458	2.117
π^3 (Wang et al., 2025c)	0.510	<u>0.729</u>	0.218	0.0999	6.912	2.036	<u>0.636</u>
GlueMap (Pan et al., 2026)	0.507	1.755	0.535	0.0984	4.337	2.308	1.388
Scal3R (Xie et al., 2026)	0.520	2.036	<u>0.215</u>	<u>0.0957</u>	2.193	<u>1.985</u>	2.589
VGGT-Long (Deng et al., 2025)	0.535	1.257	0.264	0.118	5.135	2.027	0.806
VGGT-Align (Zhang et al., 2026b)	0.536	1.258	2.141	0.118	5.109	2.027	0.808
LingBot-MAP (Chen et al., 2026a)	0.579	3.345	1.051	0.254	1.106	2.162	4.226
ZipMap (Jin et al., 2026)	0.632	1.098	1.495	0.161	11.84	2.531	1.786
VGGT (Wang et al., 2025a)	0.605	1.284	2.186	0.232	12.30	2.849	2.656
LoGeR (Zhang et al., 2026a)	0.702	1.791	2.552	0.407	2.185	3.043	3.919
TTT3R (Chen et al., 2026b)	0.711	21.19	0.441	0.305	4.798	3.457	4.638
CUT3R (Wang et al., 2025b)	0.907	10.91	6.485	0.851	4.644	2.979	4.510
9.394	4.408						
H1P3R (VGGT-Omega)	**0.468**	**0.476**	**0.154**	**0.0710**	**0.408**	**1.965**	**0.587**
H1P3R (LingBot-MAP)	0.586	2.287	0.275	0.280	<u>1.080</u>	2.209	0.668

I CHUNK AND OVERLAP EXPERIMENTS

Table 18 lists per-dataset window parameters. The main benchmark uses fixed defaults across datasets, specifically $60/30$ for VGGT-Omega and $120/32$ for LingBot-MAP. The selected windows are evaluated using frozen overlap consensus. They provide an exploratory analysis and do not enter the main ranking. All rows follow the public 5000-evaluation solver protocol. Selected versus default macro-averaged ATE means across the seven datasets are 4.403 versus $4.908\,\mathrm{m}$ for VGGT-Omega, and 3.822 versus $3.825\,\mathrm{m}$ for LingBot-MAP. Longer windows provide wider baselines for outdoor scenes, while shorter windows balance compute during rapid rotations.

Relationship between overlap consensus and the primary benchmark. In Table 18, exploratory window sweeps evaluate camera poses using frozen overlap consensus instead of solving the fine keyframe graph. Overlap consensus computes closed-form geodesic rotation and Euclidean translation averages across overlapping chunk placements. It is functionally equivalent to the `Unary`

Table 15: Per-dataset RPE_{trans} on the same seven datasets (lower is better). Bold and underline mark the full-table best and second-best values in each column.

Method	7S	ETH	KITTI	NRGBD	Oxford	T&T	VBR
VGGT-Omega (Wang et al., 2026b)	0.0172	0.101	2.358	0.0109	4.044	0.128	0.2557
π^3 (Wang et al., 2025c)	0.0158	<u>0.091</u>	<u>0.150</u>	0.0091	1.999	<u>0.106</u>	0.1591
GlueMap (Pan et al., 2026)	**0.0145**	0.583	1.805	<u>0.0079</u>	2.268	0.115	8.599
Scal3R (Xie et al., 2026)	0.0176	0.238	0.335	0.0092	<u>0.259</u>	0.107	0.2260
VGGT-Long (Deng et al., 2025)	0.0188	0.190	0.221	0.0098	0.997	0.110	0.2292
VGGT-Align (Zhang et al., 2026b)	0.0188	0.191	14.88	0.0098	1.053	0.110	0.2633
LingBot-MAP (Chen et al., 2026a)	0.0205	0.495	6.004	0.0189	0.523	0.124	2.7208
ZipMap (Jin et al., 2026)	0.0217	0.239	3.039	0.0160	3.005	0.144	0.1530
VGGT (Wang et al., 2025a)	0.0221	0.244	4.708	0.0196	5.955	0.165	0.3632
LoGeR (Zhang et al., 2026a)	0.0223	0.289	4.110	0.0180	0.547	0.182	2.5563
TTT3R (Chen et al., 2026b)	0.0263	1.229	0.329	0.0239	0.803	0.181	3.0620
CUT3R (Wang et al., 2025b)	0.0351	0.799	4.964	0.0397	1.054	0.172	3.1564
0.294	1.2625						
HIP3R (VGGT-Omega)	<u>0.0148</u>	**0.0567**	**0.0931**	**0.0049**	**0.104**	**0.103**	**0.0591**
HIP3R (LingBot-MAP)	0.0211	0.465	0.231	0.0196	0.285	0.147	<u>0.108</u>

Table 16: Dense 3D reconstruction metrics (Chamfer distance and F1 score) on VGGT-Omega versus HIP3R evaluated on datasets with dense 3D ground truth.

	Chamfer ↓		F1 ↑	
Dataset	**Baseline**	**HIP3R**	**Baseline**	**HIP3R**
ETH3D	0.0339	0.0334	98.92	99.00
Neural RGB-D	0.1310	0.0205	40.53	90.05
7-Scenes	0.0739	0.0269	58.45	87.13

pose-averaging control arm in Table 5, omitting relative motion constraints ($\mathcal{E}_f = \emptyset$). Numerical differences on individual sequences arise because the exploratory sweeps and the main benchmark converge to different coarse layout solutions on large non-convex loop graphs. On KITTI, for example, the exploratory coarse solve reached an ATE of 3.508 m, whereas the primary benchmark converged to 6.193 m. When evaluated on the identical coarse layout under strictly controlled conditions in Table 5, the complete `Shared` camera graph consistently outperforms `Unary` consensus on both backends, reducing ATE, RPE_{rot}, and RPE_{trans}.

J Ablation on the coarse-only layout

Table 19 compares camera accuracy after the coarse layout stage alone against the Full pipeline on three long sequences. The coarse layout stage primarily reduces global trajectory error (ATE). Shared-camera trajectory refinement then improves local consistency, lowering residual RPE_{rot} and RPE_{trans}. This confirms that macro layout establishes global geometry, while micro refinement enforces local kinematic smoothness.

K Controlled layout-solver comparison

To evaluate solver behavior under challenging conditions, we examine a ten-scene benchmark comprising extended outdoor traversals, including KITTI and VBR. All evaluated configurations share identical model predictions, candidate revisit pairs, and initialization. Camera refinement and point corrections are disabled to isolate layout optimization. The maximum evaluation ceiling is set to 5000 iterations. The five configurations evaluate baseline pairwise alignment, bounded point sampling, residual-fit weighting, progressive loop scheduling, and the complete optimization solver. The final step combines adaptive continuation, Huber loss, and bounded restarts.

Table 17: Dense 3D reconstruction metrics (Chamfer distance and F1 score) on LingBot-MAP versus HIP3R evaluated on datasets with dense 3D ground truth.

	Chamfer ↓		F1 ↑	
Dataset	**Baseline**	**HIP3R**	**Baseline**	**HIP3R**
ETH3D	0.1295	0.1295	86.49	86.49
Neural RGB-D	0.1620	0.0675	42.66	65.24
7-Scenes	0.1403	0.1071	54.87	72.01

Table 18: Per-dataset window settings and ATE under selected and fixed-default configurations. Bold marks a change from Table 10.

Dataset	**VGGT-Omega** (Wang et al., 2026b)			**LingBot-MAP** (Chen et al., 2026a)		
	C/O	ATE	vs $60/30$	C/O	ATE	vs $120/32$
7-Scenes	$\mathbf{90/30}$	0.0366	0.0381	$\mathbf{120/8}$	0.0801	0.0793
ETH3D	$60/30$	0.0546	0.0546	$120/32$	0.479	0.479
KITTI	$\mathbf{60/16}$	9.352	9.736	$120/32$	3.508	3.508
NRGBD	$60/30$	0.0219	0.0219	$\mathbf{160/32}$	0.0608	0.0776
Oxford	$60/30$	1.579	1.579	$120/32$	1.651	1.651
T&T	$60/30$	0.476	0.476	$120/32$	0.587	0.587
VBR	$\mathbf{120/30}$	19.30	22.45	$120/32$	20.39	20.39

We test solver robustness against noisy loop measurements by perturbing loop closures. Specifically, synthetic noise is injected into 25% of detected loop edges using a fixed random seed. For each selected edge, we apply a 10° rotation error. We also inject a 25% translation offset and a 0.10 log-scale distortion. The evaluation confirms efficiency gains from bounded sampling, alongside accuracy and runtime gains from the complete solver. The complete configuration reduces clean mean ATE by 19.77% relative to the baseline. This improvement concentrates in VBR, where mean error falls from 28.6017 to 22.4461 m. Peak host memory remains comparable across configurations.

L PLACEMENT AND MOTION CONTROL PROTOCOL

The four trajectory refinement configurations in Table 5 evaluate different strategies for resolving multiple camera predictions under identical coarse layouts. Layout Only is the coarse-stage first-writer export without shared-camera trajectory refinement. Single keeps one prediction per frame from the recovered layout. Unary performs robust pose averaging over overlapping observations without relative-motion factors. Motion optimizes within-chunk relative-motion factors without placement priors. Shared jointly optimizes placement priors and relative-motion constraints as formulated in Eq. (6). All trajectory-refinement configurations are evaluated under identical protocols and macro-averaged across all benchmarks for direct comparison. Shared improves aggregate pose AUC thresholds and error metrics over Single and Unary. Both Layout Only and Single use identical coarse placements across benchmarks. These results show that shared-camera trajectory refinement preserves global layout consistency while improving local pose estimates.

M DENSE GEOMETRY COMPARISON AND KEYFRAME ANALYSIS

Table 6 in the main text reports stage-wise trajectory results from coarse layout recovery to fine trajectory refinement. This appendix extends that comparison to dense surface geometry, keyframe subgraphs, and window settings. The coarse stage determines relative scale and placement of chunks from overlap and loop constraints. Holding $A^\star$ fixed keeps chunk transforms out of frame-level optimization. Local camera adjustments cannot alter chunk scales. Placement priors keep refined cameras on this layout while allowing relative poses to improve. On extended sequences like VBR, coarse layout optimization eliminates cumulative scale drift across chunks. Shared-camera trajectory refinement then focuses on local relative precision, reducing $\mathrm{RPE}_{\mathrm{trans}}$ by 22%.

Table 19: Trajectory accuracy of Layout Only versus Full on the same feed-forward predictions.

		Layout Only			Full		
Sequence	T	ATE	RPE_{rot}	$\text{RPE}_{\text{trans}}$	ATE	RPE_{rot}	$\text{RPE}_{\text{trans}}$
KITTI seq00	4541	9.750	0.202	0.131	9.730	0.154	0.093
spagna_train0	14,141	10.256	0.542	0.048	10.261	0.535	0.034
campus_train1	11,671	7.381	0.230	0.076	7.387	0.157	0.045

Table 20: Ablation of layout-solver components on extended sequences. Values are arithmetic means over the ten long-scene evaluation set. Total time includes measurement construction and all layout solves. Final solve is the duration of the final least-squares call.

Configuration	ATE	RPE_{rot}	$\text{RPE}_{\text{trans}}$	Graph Build & Solve (s)	Final solve (s)
Clean loop measurements					
Style control	21.7751	1.31094	0.11296	12232.18	1404.54
+ Bounded sampling	21.7569	1.31093	0.11280	2004.20	1345.43
+ Fit weighting	21.7770	1.31084	0.11262	2077.49	1387.14
+ Progressive loop scheduling	21.5241	1.31077	0.11070	1477.59	688.02
Complete	17.4708	1.30972	0.09834	753.43	41.28
Perturbed loop measurements					
Style control	20.4767	1.33444	0.11078	9343.00	1226.62
+ Bounded sampling	20.5269	1.33447	0.11109	1641.88	1220.56
+ Fit weighting	21.2226	1.33061	0.11627	1798.38	1284.72
+ Progressive loop scheduling	19.1915	1.32976	0.10568	1032.27	461.17
Complete	16.9298	1.32951	0.10280	794.13	31.31

Surface geometry comparison across stages. Table 21 evaluates dense 3D reconstruction quality across ETH3D, Neural RGB-D, and 7-Scenes under three output configurations. Native evaluates unaligned chunk predictions concatenated in overlapping coordinates. Layout Only evaluates globally placed chunks without shared-camera trajectory refinement. Full then uses one shared camera per frame for those maps. The Chamfer drop from Native to Layout Only is the layout effect. Full mainly changes which cameras see the already placed points.

Table 21: Matched geometry comparison. Layout Only removes shared-camera frame corrections. F1 is reported in percent. Differences in F1 are percentage points. These are geometry-subset results, not KITTI or Oxford geometry evaluations.

Backend	Output	Chamfer ↓	F1 ↑
VGGT-Omega	Native	0.075855	65.918454
VGGT-Omega	Layout Only	0.027289	91.196090
VGGT-Omega	Full	0.027140	91.292886
LingBot-MAP	Native	0.142331	61.131371
LingBot-MAP	Layout Only	0.104866	74.492888
LingBot-MAP	Full	0.104187	74.599068

Relative to the layout-only export, Full reduces mean Chamfer by 0.55% with VGGT-Omega and 0.65% with LingBot-MAP. F1 increases by 0.097 and 0.106 percentage points, respectively. These results indicate a clear division of roles between the two stages. Global layout alignment establishes large-scale surface consistency, while shared-camera trajectory refinement further adjusts viewpoint poses without modifying the predicted feed-forward surface depth.

Keyframe efficiency. On long videos, a keyframe subgraph reduces the number of camera variables, while interpolated corrections recover poses for the remaining frames. Appendix C evaluates this trade-off in Table 9. On the 4541-frame KITTI sequence, our stride-5 keyframe subgraph optimizes 1059 variables and solves in 75 ms on CPU. Its ATE closely matches the dense graph (9.730 vs. 9.731 m). Relative precision changes only slightly, with RPE_{rot} at 0.154° versus 0.144° and $\text{RPE}_{\text{trans}}$ at 0.093 versus 0.086. We interpolate rotation and translation corrections Δ_t and apply them to the

corresponding per-frame poses. This preserves each frame's local prediction as the baseline rather than replacing the trajectory with interpolated keyframe poses.

Chunk sizes and overlaps. Table 18 in Appendix I compares selected window configurations against default settings. Longer VGGT-Omega windows on VBR further improve ATE, while performance across other settings remains generally stable. The impact of window selection depends on scene structure and backend. The main comparison uses one window setting per backend. Recorded solver settings and alignment parameters appear in Appendix D.