

Anchor3R: Streaming 3D Reconstruction with Transient Anchors for Long-Horizon Visual Mapping

Peilin Tao^{1,2,3} Chong Cheng^{3,4} Yuansen Du^{3,†} Caiwei Song³ Zhengqing Chen³
Xiaoyang Guo³ Wei Yin³ Weiqiang Ren³ Qian Zhang³ Hainan Cui^{1,2,†} Shuhan Shen^{1,2,†}

¹Institute of Automation, Chinese Academy of Sciences

²School of Artificial Intelligence, University of Chinese Academy of Sciences

³Horizon Robotics ⁴HKUST(GZ)

[†] Corresponding author [‡] Project lead



Figure 1: **Anchor3R results on campus-scale sequences.** We visualize our reconstruction on campus_train0 and campus_train1 of the VBR dataset, two real-world sequences with **12,042** and **11,671** frames. Predicted trajectories are color-coded and ground truth is shown as white curves.

Abstract: Long-horizon online visual mapping requires continuous camera-motion and scene-geometry estimation under bounded computation. Recent feed-forward 3D reconstruction models provide strong geometric priors, but streaming variants often predict poses in a fixed or historically maintained coordinate system, leading to train–test mismatch, early-anchor attention bias, and accumulated drift. We propose *Anchor3R*, a current-centric streaming 3D reconstruction framework that predicts window-relative poses and local geometry in the current-frame coordinate system. Overlapping predictions form a dense relative-pose graph, supporting online pose updates and loop-aware motion averaging for global reconstruction. Experiments on indoor, outdoor, driving, and RGB-D benchmarks demonstrate improved long-horizon pose accuracy and dense reconstruction quality over existing streaming baselines. Despite being trained only on 48-frame sequences, Anchor3R directly generalizes to streams exceeding 10,000 frames while maintaining bounded GPU memory during online inference. Code is available at <https://github.com/polar-explorer/Anchor3R>.

Keywords: Robot Visual Mapping, Streaming Feed-forward 3D Reconstruction

1 Introduction

Although recent streaming variants improve scalability through recurrent scene memories [1, 2, 3], causal Transformer caches [4, 5, 6], or compact camera/state pools [7, 8], most of them still formulate streaming reconstruction as sequential state prediction in a fixed or historically maintained gauge. This forces the model to propagate a long-lived coordinate system through hidden states, entangling local geometric reasoning with historical gauge maintenance. Such coupling can lead to attention sinks around early anchors [6] and causes uncertainty and scale errors to accumulate over long sequences. Moreover, the accumulated outputs form a trajectory rather than a graph of independent visual measurements, making global error redistribution difficult. This raises a key question: *how should feed-forward 3D models expose their predictions for large-scale streaming reconstruction?* We argue that streaming mapping should decouple local measurement from global gauge alignment.

We propose *Anchor3R*, a current-centric streaming 3D reconstruction framework for long-horizon visual mapping. Instead of maintaining a persistent global gauge, Anchor3R uses the current frame as a transient anchor and directly predicts window-relative pose measurements $\{\hat{\mathbf{T}}_{i \leftarrow t}\}_{i \in \mathcal{W}_t}$. This converts streaming reconstruction from fixed-gauge state regression into repeated local measurement prediction, allowing the network to focus on short-range relative geometry reasoning without propagating long-range gauge drift through attention or memory. As the window advances, overlapping predictions constrain each frame from multiple anchors and naturally form a dense relative-pose graph, which supports online pose updates and loop-aware motion averaging.

To realize this formulation, Anchor3R introduces a pose-query-based streaming Transformer that separates frame-level visual evidence from window-level pose reasoning. Decoupled frame attention computes image-conditioned states that can be reused across windows, while current-centric window attention instantiates pose-query tokens for the active window and reasons under the current-frame gauge. We cache only image-side key/value states and discard pose-query states after each local prediction. This image-only cache stores gauge-agnostic visual and correspondence cues, while preventing pose tokens tied to one transient gauge from being propagated into later windows. Therefore, Anchor3R achieves bounded GPU-memory streaming inference without explicit cache refresh.

Finally, Anchor3R accumulates window-relative pose measurements into an explicit motion graph. In the online mode, the current pose is estimated from multiple measurements connected to previously recovered frames. For offline refinement, retrieved loop-closure keyframes are reinserted as transient anchors to generate long-range relative-pose edges, and motion averaging redistributes drift over the graph. The recovered global poses then align local point maps into a coherent reconstruction. Thus, the offline refinement is a direct consequence of the proposed prediction interface: Anchor3R operates on a dense relative-pose graph rather than a single predicted trajectory.

Our contributions are threefold: 1) We formulate streaming feed-forward reconstruction as current-centric dense relative-pose prediction, replacing persistent global-gauge regression with window-relative predictions anchored at the current frame. 2) We design an image-only cached pose-query Transformer that separates reusable frame evidence from transient current-gauge pose reasoning, enabling bounded GPU-memory streaming inference without propagating stale pose states. 3) We accumulate dense relative-pose measurements into a motion graph that supports online pose updates and loop-aware motion averaging, enabling global drift redistribution from overlapping local predictions.

2 Related Work

Classical SfM and SLAM. Classical SfM and SLAM systems remain strong baselines for visual mapping because they explicitly enforce multiview geometric constraints. Incremental SfM [9] estimates pairwise relative poses from image correspondences and sequentially registers cameras using Perspective- n -Point estimation [10], triangulation [11], and bundle adjustment [12, 13]. SLAM systems [14, 15] further emphasize online tracking, local mapping, loop closure, and map refinement. Global SfM [16] instead estimates camera motions jointly from pairwise relative poses via rotation

and translation averaging [17, 18, 19], followed by triangulation and refinement. Such local-to-global decomposition is attractive for long-horizon mapping as it separates relative measurement estimation from global alignment. Anchor3R follows the same principle, but replaces correspondence-based relative pose estimation with dense relative-pose prediction from a streaming feed-forward model.

Offline Feed-forward Reconstruction. Recent feed-forward visual geometry models aim to recover camera motion and dense structure directly from RGB inputs. DUS3R [20] introduced end-to-end two-view pointmap prediction in a shared pairwise coordinate frame. VGGT [21] further unified camera, depth, point map, and track prediction within a Transformer, while π^3 [22] removed fixed-reference bias through a permutation-equivariant formulation. More recent models extend this paradigm to broader settings and larger scales, including Depth Anything 3 [23], MapAnything [24], and VGG-T³ [25]. Several works further improve scalability by processing long sequences or image collections through chunks, compact states, or test-time adaptation. VGGT-Long [26] uses chunk-wise reconstruction, overlap-based alignment, and loop closure; ZipMap [27] compresses an image collection into a compact hidden scene state with test-time training layers; LoGeR [28] combines hybrid memory with chunked long-video reconstruction; and Scal3R [29] introduces test-time-adapted global context for large-scale reconstruction. These methods provide powerful geometric priors and improve long-context scalability, but they are mainly designed for offline or chunk-based processing over fixed inputs. In contrast, Anchor3R targets streaming visual mapping, where the model must output local pose and geometry updates as frames arrive while still supporting global error redistribution over the accumulated motion graph.

Streaming Feed-forward Reconstruction. Native streaming reconstruction processes frames incrementally and must maintain useful historical information under bounded or partially bounded memory. CUT3R [1] uses persistent latent scene memory to support continuous prediction, while Point3R [30] introduces explicit spatial pointer memory anchored to reconstructed 3D structure. SStream3R [4] formulates sequential reconstruction with a causal Transformer, and WinT3R [7] combines sliding-window processing with a compact camera token pool. LongStream [6] identifies first-frame anchoring as a major source of long-sequence degradation and predicts keyframe-relative poses to reduce gauge coupling. Recent methods such as TTT3R [2] and Mem3R [8] further explore inference-time state updating and hybrid memory mechanisms. Despite these advances, many streaming methods still rely on the backbone memory or cache to implicitly maintain a coordinate gauge over time. As a result, local geometric prediction, historical state propagation, and long-range consistency become tightly coupled, making it difficult for later observations to correct accumulated drift. In contrast, Anchor3R decouples scalable bounded-window prediction from global consistency recovery: the network predicts current-centric window-relative poses as local measurements, and motion averaging integrates them into a globally consistent trajectory.

3 Method

Anchor3R adopts a local-to-global design for long-horizon streaming reconstruction. It uses the current frame as a transient anchor to predict relative-pose measurements within an active window, while local pointmaps are predicted in the current-frame coordinate system. Across overlapping windows, these measurements form a dense relative-pose graph, which supports online pose updates and loop-aware motion averaging. Thus, neural prediction focuses on short-range relative geometry, while graph optimization handles long-range gauge alignment and drift redistribution.

3.1 Current-Centric Formulation

Given an image stream $\mathcal{I} = \{I_1, \dots, I_T\}$, Anchor3R processes the sequence with an active sliding window $\mathcal{W}_t = \{k, \dots, t\}$ at time t , where $k = \max(1, t - W + 1)$. For each frame $i \in \mathcal{W}_t$, the network predicts a relative pose $\hat{\mathbf{T}}_{i \leftarrow t} \in \text{SE}(3)$ from the current frame I_t to frame I_i . Given global camera poses $\mathbf{T}_i \in \text{SE}(3)$, the training target is

$$\mathbf{T}_{i \leftarrow t} = \mathbf{T}_i \mathbf{T}_t^{-1}, \quad \mathbf{T}_{t \leftarrow t} = \mathbf{I}. \quad (1)$$

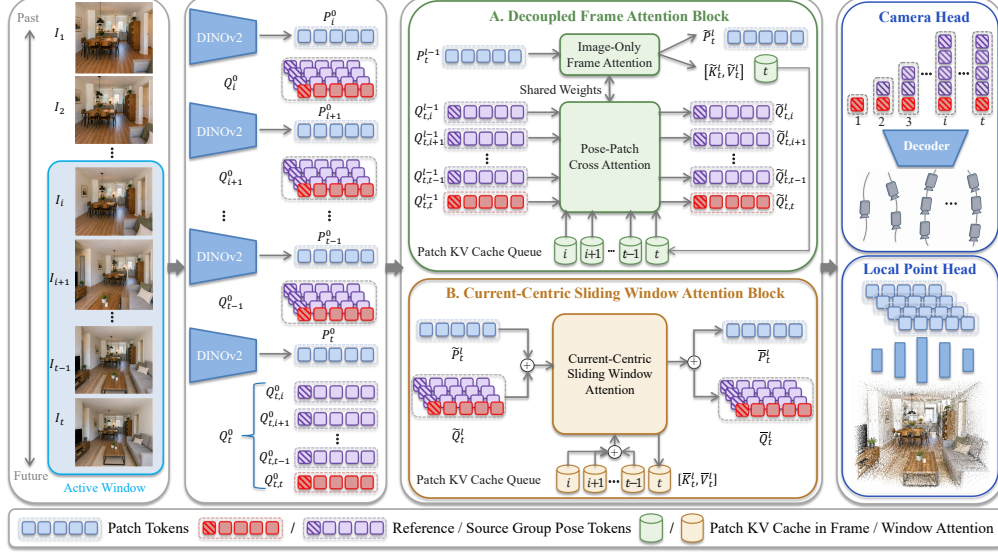


Figure 2: **Overview of Anchor3R.** Given the current frame I_t , Anchor3R extracts DINOv2 patch tokens and instantiates grouped pose-query tokens for frames in $\mathcal{W}_t$, using I_t as the local reference. A sliding-window pose-query Transformer alternates between decoupled frame attention and current-centric window attention. The camera head decodes window-relative poses, while the point head predicts the current-frame pointmap.

This target depends only on the relative geometry within the active window, rather than on an increasingly distant global origin. Therefore, the network can focus on local visual overlap, correspondence reasoning, and short-range geometric consistency, without propagating a long-lived gauge through hidden states or attention caches. In parallel, Anchor3R predicts a current-frame local pointmap $\hat{\mathbf{X}}_t^{\text{local}}$. At time t , the prediction produces a set of graph edges $\mathcal{E}_t = \{(i, t, \hat{\mathbf{T}}_{i \leftarrow t}) \mid i \in \mathcal{W}_t, i < t\}$. As the window slides, repeated observations from later anchors add redundant edges for the same frames, turning local window predictions into a dense relative-pose measurement graph. The recovered global poses then transform local pointmaps into a coherent reconstruction.

3.2 Sliding-Window Pose-Query Transformer

For each incoming frame I_t , a DINOv2 encoder extracts patch tokens $\mathbf{P}_t^0 \in \mathbb{R}^{N \times C}$, while historical frames provide cached image-conditioned states. For each frame $i \in \mathcal{W}_t$, we instantiate a pose-query group $\mathbf{Q}_{t,i}^0 \in \mathbb{R}^{(1+X) \times C}$, consisting of one pose token and X register tokens. The full query set is $\mathbf{Q}_t^0 = \text{Concat}_{i \in \mathcal{W}_t} \mathbf{Q}_{t,i}^0$, where the current frame uses a learnable reference template $\mathbf{Q}^{\text{ref}}$ and source frames use $\mathbf{Q}^{\text{src}}$. This asymmetric initialization defines the current frame as the local reference and lets source queries aggregate motion evidence relative to it.

At layer l , the **decoupled frame attention** block updates only the current image tokens, $(\tilde{\mathbf{P}}_t^l, \tilde{\mathbf{K}}_t^l, \tilde{\mathbf{V}}_t^l) = \mathcal{F}^l(\mathbf{P}_t^{l-1})$, and caches the image-side keys and values. Each pose-query group then reads from its corresponding frame representation, $\tilde{\mathbf{Q}}_{t,i}^l = \mathcal{F}^l(\mathbf{Q}_{t,i}^{l-1}; \tilde{\mathbf{K}}_i^l, \tilde{\mathbf{V}}_i^l)$, where historical image states are reused from the cache. Thus, frame-level visual evidence is computed once and remains independent of future pose-query instantiations. The **current-centric window attention** block jointly updates the current patch tokens and all pose-query groups: $(\mathbf{P}_t^l, \{\mathbf{Q}_{t,i}^l\}_{i \in \mathcal{W}_t}, \bar{\mathbf{K}}_t^l, \bar{\mathbf{V}}_t^l) = \mathcal{W}^l(\text{Concat}(\tilde{\mathbf{P}}_t^l, \{\tilde{\mathbf{Q}}_{t,i}^l\}_{i \in \mathcal{W}_t}); \bar{\mathbf{K}}_{\mathcal{W}_t}^l, \bar{\mathbf{V}}_{\mathcal{W}_t}^l)$, where $\bar{\mathbf{K}}_{\mathcal{W}_t}^l$ and $\bar{\mathbf{V}}_{\mathcal{W}_t}^l$ concatenate cached image-side states from frames in $\mathcal{W}_t \setminus \{t\}$. This design separates reusable visual evidence from transient pose reasoning: pose-query tokens are tied to a current-centric gauge, so caching them across windows would mix stale coordinate hypotheses from different anchors. In contrast, image-conditioned patch states store gauge-agnostic appearance, geometry, and matching cues. By caching only image-side states, Anchor3R keeps correspondence information in patch tokens while rebuilding gauge-dependent pose reasoning for each window.

3.3 Prediction Heads

After the final layer, Anchor3R predicts window-relative poses and current-frame local geometry. For each query group $\mathbf{Q}_{t,i}^L$, its first token is used as the pose token, and the camera head jointly decodes $\{\hat{\mathbf{T}}_{i \leftarrow t}\}_{i \in \mathcal{W}_t} = \mathcal{H}_{\text{cam}}(\text{Concat}_{i \in \mathcal{W}_t} \mathbf{Q}_{t,i}^L[1])$. Joint decoding is used because all relative poses in the active window share the same current-frame reference. In parallel, a DPT-style point head predicts $[\hat{\mathbf{X}}_t^{\text{local}}, \hat{\Sigma}_t] = \mathcal{H}_{\text{pt}}(\{\mathbf{P}_t^l\}_{l \in \mathcal{L}}) \in \mathbb{R}^{H_0 \times W_0 \times 4}$ from selected multi-level patch features, where $\hat{\mathbf{X}}_t^{\text{local}}$ is the current-frame pointmap and $\hat{\Sigma}_t$ is the confidence map in the local camera coordinate system.

3.4 Motion Graph and Pose Recovery

The predicted relative poses define a motion graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where vertices are frames and edges store window-relative measurements. Let the global pose of frame i be represented by rotation $\mathbf{R}_i$ and camera center $\mathbf{c}_i$. For an edge (i, t) , the network predicts $(\hat{\mathbf{R}}_{i \leftarrow t}, \hat{\mathbf{t}}_{i \leftarrow t})$, with $\hat{\mathbf{R}}_{i \leftarrow t} \approx \mathbf{R}_i \mathbf{R}_t^\top$ and $\hat{\mathbf{v}}_{i,t} = \hat{\mathbf{R}}_{i \leftarrow t}^\top \hat{\mathbf{t}}_{i \leftarrow t} \approx \mathbf{c}_t - \mathbf{c}_i$. Instead of composing poses along a spanning tree, we recover global poses by averaging over the full graph, which exploits redundant local predictions and redistributes errors across the trajectory. We fix the first-frame gauge by setting $\hat{\mathbf{R}}_1 = \mathbf{I}$ and $\hat{\mathbf{c}}_1 = \mathbf{0}$.

For online inference at time t , Anchor3R estimates the current global pose from the $|\mathcal{W}_t| - 1$ newly predicted relative poses $\{\hat{\mathbf{T}}_{i \leftarrow t}\}_{i \in \mathcal{W}_t \setminus \{t\}}$ and the previously estimated global poses $\{(\hat{\mathbf{R}}_i, \hat{\mathbf{c}}_i)\}_{i \in \mathcal{W}_t \setminus \{t\}}$. Each historical frame induces a candidate current rotation and center; we take the Lie-algebra median of the candidate rotations for $\hat{\mathbf{R}}_t$ and the coordinate-wise median of candidate centers $\hat{\mathbf{c}}_i + \hat{\mathbf{v}}_{i,t}$ for $\hat{\mathbf{c}}_t$. For offline refinement, we jointly optimize all rotations and centers:

$$\min_{\{\mathbf{R}_i\}} \sum_{(i,t) \in \mathcal{E}} \rho_R \left(\left\| \text{Log}(\hat{\mathbf{R}}_{i \leftarrow t}^\top \mathbf{R}_i \mathbf{R}_t^\top) \right\|_2 \right), \quad \min_{\{\mathbf{c}_i\}} \sum_{(i,t) \in \mathcal{E}} \|\mathbf{c}_i - \mathbf{c}_t + \hat{\mathbf{v}}_{i,t}\|_1. \quad (2)$$

The rotation objective is solved by IRLS in the Lie algebra, and the translation objective is solved by ADMM [31]. Historical or loop-closure keyframes can be reinserted into the active window to add long-range edges to $\mathcal{G}$, enabling global error redistribution beyond local streaming updates.

3.5 Training

Training objectives. Following VGGT [21], we train Anchor3R with a multi-task objective $\mathcal{L} = \lambda_{\text{cam}} \mathcal{L}_{\text{cam}} + \mathcal{L}_{\text{pmap}}$, where λ_{cam} balances the camera and point-map losses. The camera loss supervises the relative camera parameters predicted within each active window: $\mathcal{L}_{\text{cam}} = \sum_t \sum_{i \in \mathcal{W}_t} (\|q_{i \leftarrow t} - \hat{q}_{i \leftarrow t}\|_1 + \|\mathbf{t}_{i \leftarrow t} - s^* \hat{\mathbf{t}}_{i \leftarrow t}\|_1)$, where $q_{i \leftarrow t}$ and $\mathbf{t}_{i \leftarrow t}$ denote the ground-truth relative rotation and translation, respectively. The local point-map loss combines confidence-weighted reconstruction with gradient regularization: $\mathcal{L}_{\text{pmap}} = \sum_t \|\hat{\Sigma}_t \odot (\mathbf{X}_t^{\text{local}} - s^* \hat{\mathbf{X}}_t^{\text{local}})\| + \|\hat{\Sigma}_t \odot (\nabla \mathbf{X}_t^{\text{local}} - \nabla(s^* \hat{\mathbf{X}}_t^{\text{local}}))\| - \alpha \log \hat{\Sigma}_t$, where $\hat{\Sigma}_t$ is the predicted confidence map. The scale factor is estimated following π^3 [22] as $s^* = \arg \min_s \sum_t \|\mathbf{X}_t^{\text{local}} - s \hat{\mathbf{X}}_t^{\text{local}}\|_1$. We apply the same s^* to both relative camera translations and local point maps, encouraging the predicted cameras and geometry to remain scale-consistent within each active window and across overlapping windows.

Training data. We train Anchor3R on a mixture of real and synthetic datasets, including Wil-dRGB [32], ScanNet [33], HyperSim [34], Mapillary [35], Replica [36], Mapfree [37], TartanAir [38], MVS-Synth [39], Virtual KITTI [40], Aria Synthetic Environments [41], Spring [42], Waymo Open [43], BlendedMVS [44], Co3Dv2 [45], MegaDepth [46] and DL3DV [47]. For unordered data, we construct overlapping sequences with pose-guided sampling; for video data, we use random interval sampling and block shuffling to increase diversity while preserving local continuity.

Implementation details. Anchor3R is initialized from VGGT and retains its 24-layer backbone with alternating attention block, resulting in approximately 1.2B parameters. During training, we unroll each sample over 48 frames, while using a fixed active window of $W = 10$ frames for local prediction. Each pose-query group contains one pose token and 31 register tokens. The model is

Table 1: **Quantitative comparison on KITTI [48] in terms of ATE.** The upper block lists optimization-based baselines, and the lower block reports streaming feed-forward methods. Failed cases are marked by * and excluded from the average. Anchor3R achieves the best average accuracy.

Methods	KITTI [48] (ATE ↓)											Avg.
	00 4541 ×, 3.7 km	01 1101 ×, 2.5 km	02 4661 ×, 5.1 km	03 801 ×, 0.6 km	04 271 ×, 0.4 km	05 2761 ×, 2.2 km	06 1101 ×, 1.2 km	07 1101 ×, 0.7 km	08 4071 ×, 3.2 km	09 1591 ×, 1.7 km	10 1201 ×, 0.9 km	
FastVGGT	*	705.39	*	62.38	10.27	157.74	124.43	69.27	*	190.10	194.75	189.29
MASt3R-SLAM	*	530.37	*	18.87	88.98	159.430	92.00	*	*	*	*	177.93
VGGT-SLAM	*	607.16	*	169.83	13.12	*	*	*	*	*	*	263.37
VGGT-Long	8.64	21.20	52.72	8.78	4.20	<u>9.88</u>	4.67	2.66	72.98	31.84	27.71	<u>25.94</u>
CUT3R	185.89	651.52	296.98	148.06	22.17	155.61	132.54	77.03	238.39	205.94	193.39	209.78
TTT3R	190.93	546.84	218.77	105.28	11.62	153.12	132.94	70.95	180.57	211.01	133.00	177.73
STream3R	190.98	681.95	301.40	158.25	102.73	159.85	135.03	90.37	261.15	216.31	207.49	227.77
StreamVGGT	191.93	653.06	303.35	157.50	108.24	160.46	133.71	89.00	263.95	216.69	209.80	226.15
LongStream	92.55	46.01	134.70	3.81	<u>1.95</u>	84.69	23.12	14.93	62.07	85.61	21.48	51.90
Anchor3R-Online	44.18	48.43	149.61	4.39	1.99	62.63	12.12	10.39	<u>52.12</u>	55.23	<u>8.57</u>	40.89
Anchor3R-Offline	<u>19.68</u>	48.62	<u>75.75</u>	<u>4.22</u>	1.90	7.56	<u>6.36</u>	<u>7.63</u>	49.34	<u>49.02</u>	8.00	25.03

Table 2: **Quantitative comparison on VBR [49].** We report ATE, where lower is better.

Method	VBR [49] ATE ↓							Avg.
	campus_train0 12042 ×, 2.73 km	campus_train1 11671 ×, 2.95 km	ciampino_train1 18846 ×, 5.20 km	colosseo_train0 8815 ×, 1.45 km	diag_train0 10021 ×, 1.02 km	pincio_train0 11142 ×, 1.27 km	spagna_train0 14141 ×, 1.56 km	
VGGT-SLAM	93.51	71.74	124.10	101.00	33.64	66.42	57.00	78.20
VGGT-Long	118.59	98.21	172.13	<u>39.56</u>	30.80	53.44	50.27	80.43
Pi3-Chunk	<u>78.50</u>	<u>65.77</u>	<u>111.72</u>	<u>77.09</u>	<u>23.81</u>	<u>41.99</u>	<u>44.76</u>	63.38
InfiniteVGGT	123.65	100.00	*	83.91	31.58	70.73	56.25	91.60
LongStream	100.57	105.55	131.78	72.52	32.35	43.47	59.31	77.93
Anchor3R-Online	86.63	82.16	168.25	61.43	29.38	51.34	54.34	76.21
Anchor3R-Offline	5.13	3.52	78.64	17.25	5.35	15.56	11.76	19.60

optimized with AdamW using cosine learning-rate decay, a peak learning rate of 1×10^{-4} , and 2k warm-up steps. Images are resized such that their longer side is at most 518 pixels, with aspect-ratio jittering applied for data augmentation. Training is performed for 80k iterations on 32 NVIDIA A800 GPUs and takes approximately 12 days.

4 Experiments

We evaluate Anchor3R for long-horizon robot visual mapping through one-pass streaming pose estimation, loop-aware graph refinement, dense reconstruction, and controlled analyses of its current-centric formulation and streaming scalability. For camera pose estimation, we use KITTI Odometry [48], VBR [49], TUM RGB-D [50], Oxford Spires [51], and Waymo [43], covering driving, handheld/vehicle mapping, indoor RGB-D, and mobile mapping trajectories. All main evaluation sequences are excluded from training; for Waymo, we use a held-out subset of the official training split. For dense reconstruction, we evaluate on 7Scenes [52] and TUM RGB-D. We further conduct controlled ablations on Virtual KITTI [40] with a ViT-Small backbone.

4.1 Camera Pose Estimation

Evaluation protocol. We align each predicted trajectory to the ground truth with a similarity transformation and report Absolute Trajectory Error (ATE), where lower is better. Failed cases are marked by * and excluded from the average. We compare against optimization-based systems, including FastVGGT [53], MASt3R-SLAM [54], VGGT-SLAM [55], VGGT-Long [26] and Pi3-Chunk [22], as well as streaming feed-forward methods, including CUT3R [1], TTT3R [2], STream3R [4], StreamVGGT [5], LongStream [6], and InfiniteVGGT [56]. We report two variants. Anchor3R-Online follows a strict one-pass streaming protocol: frames are processed in temporal order, poses are incrementally updated from current-centric window measurements, and no loop detection or loop-keyframe reinsertion is used. Anchor3R-Offline evaluates the graph-refinement capability enabled by our prediction interface. It augments the accumulated relative-pose graph with loop-closure measurements and performs motion averaging. Specifically, we select keyframes every 5 frames, retrieve the top-3 similar images using NetVLAD [57], and reinsert temporally separated matches as transient anchors to predict long-range relative-pose measurements.

Table 3: **Quantitative comparison on TUM, Oxford Spires, and Waymo.** Top: optimization-based methods; Bottom: streaming methods. Anchor3R remains robust across short indoor trajectories and mobile mapping sequences.

Methods	TUM [50]	Oxford Spires [51]	Waymo [43]
	ATE ↓	ATE ↓	ATE ↓
FastVGGT	0.418	36.577	1.281
MASt3R-SLAM	0.082	37.728	7.625
VGGT-SLAM	0.123	31.003	7.431
CUT3R	0.542	32.440	9.396
TTT3R	0.308	36.214	3.486
STream3R	0.633	37.569	42.203
StreamVGGT	0.627	37.255	45.101
LongStream	0.076	19.815	0.737
Anchor3R-Online	0.091	17.661	0.425

Table 4: **Quantitative comparison on 7Scenes and TUM.** CD ↓ and F1@0.25 ↑ are adopted for evaluation. Best numbers are in bold; second-best numbers are underlined.

Methods	7Scenes		TUM	
	CD ↓	F1@0.25 ↑	CD ↓	F1@0.25 ↑
FastVGGT	6.373	0.710	<u>0.104</u>	0.926
MASt3R-SLAM	5.987	0.691	0.057	0.954
VGGT-SLAM	6.306	0.696	1.993	0.633
CUT3R	6.281	0.274	0.474	0.533
TTT3R	6.231	0.260	0.249	0.792
STream3R	6.353	0.479	1.126	0.444
StreamVGGT	6.630	0.483	0.680	0.402
LongStream	<u>2.260</u>	0.641	0.225	0.673
Anchor3R-Online	1.848	<u>0.707</u>	0.108	<u>0.933</u>

Results. Table 1 reports results on KITTI. Existing streaming feed-forward methods still show large errors on kilometer-scale trajectories, suggesting that recurrent states or causal caches alone are insufficient for stable long-horizon pose estimation. Under the strict online protocol without loop closure, Anchor3R-Online reduces the average ATE from 51.90 m for LongStream to 40.89 m, demonstrating the benefit of current-centric local measurement prediction independently of offline refinement. Anchor3R-Offline further reduces the error to 25.03 m by optimizing the accumulated dense relative-pose graph. Table 2 evaluates longer and more diverse real-world routes on VBR. Anchor3R-Online remains competitive with prior streaming methods, while Anchor3R-Offline achieves the best average ATE and ranks first on all seven sequences among the methods in Table 2. The larger Online–Offline improvement on VBR is consistent with the availability of long-range revisits: most VBR trajectories contain loop closures, whereas only some KITTI sequences do. This suggests that current-centric local measurements are useful both for strict online estimation and as reusable constraints for later global refinement. Table 3 further demonstrates generalization across indoor RGB-D, mobile mapping, and driving scenarios.

4.2 3D Reconstruction

Evaluation protocol. We evaluate dense reconstruction on 7Scenes [52] and TUM RGB-D [50]. For each method, we reconstruct a point cloud from the predicted camera trajectory and depth or point-map outputs, align it to the ground truth with a similarity transformation, and report Chamfer Distance (CD) and F1 score at a threshold of 0.25.

Results. Table 4 compares online reconstruction quality. Anchor3R-Online achieves the best CD on 7Scenes and the second-best F1 score, substantially improving over streaming baselines such as CUT3R [1], TTT3R [2], STream3R [4], and StreamVGGT [5]. On TUM, MASt3R-SLAM [54] performs best due to SLAM-style geometric optimization on short indoor trajectories, while Anchor3R-Online remains competitive with the second-best F1 score. These results show that even without loop-aware offline refinement, current-centric streaming pose updates provide sufficiently stable camera estimates to align local pointmaps into coherent reconstructions.

4.3 Ablation and Analysis

We further conduct controlled analyses to study the current-centric formulation, cache design, active-window size, streaming scalability, and the effect of offline refinement. Unless otherwise specified, diagnostic ablations are conducted on Virtual KITTI [40] with a ViT-Small backbone.

Current-centric prediction. A STream3R-like fixed-gauge streaming baseline [4] implicitly uses the first frame as a persistent global anchor. As observed in LongStream [6], such first-frame anchoring can create an attention sink, especially when later frames become weakly overlapping with the initial view. Figure 3 visualizes attention over 60 vKITTI frames. The first-centric baseline retains strong attention to the first frame throughout the sequence, forming a persistent first-frame

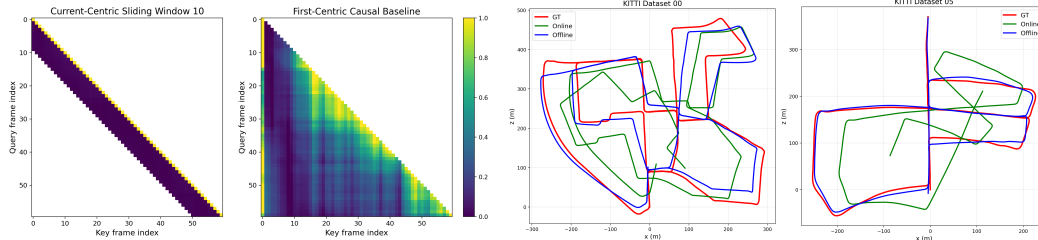


Figure 3: **Attention-score comparison.** The first-centric baseline exhibits persistent attention to the first frame throughout the sequence, while Anchor3R shifts its attention support forward with the current-centric sliding window. Figure 4: **Qualitative comparison on KITTI.** Loop-closure constraints make the offline trajectories of sequences 00 and 05 more globally consistent and closer to the ground truth.

Table 5: **Quantitative comparison on vKITTI.** For each scene, we report ATE, relative rotation error (RRE), and relative translation error (RTE). The number of images and trajectory length are shown under each scene name. Best results are highlighted in bold.

Methods	Scene 01 447×, 332 m			Scene 02 223×, 113 m			Scene 06 270×, 51 m			Scene 18 339×, 254 m			Scene 20 837×, 711 m			Avg.		
	ATE ↓	RTE ↓	RRE ↓	ATE ↓	RTE ↓	RRE ↓	ATE ↓	RTE ↓	RRE ↓	ATE ↓	RTE ↓	RRE ↓	ATE ↓	RTE ↓	RRE ↓	ATE ↓	RTE ↓	RRE ↓
w pose cache	3.613	0.177	0.132	2.174	0.111	0.074	0.285	0.037	0.055	3.955	0.195	0.087	7.558	0.211	0.112	3.517	0.146	0.092
w/o pose cache	3.113	0.101	0.127	1.054	0.084	0.073	0.424	0.042	0.059	1.410	0.118	0.080	7.208	0.142	0.110	2.642	0.097	0.090

attention pattern even as the query advances, while Anchor3R shifts its attention support forward with the current-centric sliding window. This result supports our first contribution: replacing persistent global-gauge regression with current-centric local measurement prediction reduces the burden of long-range coordinate maintenance inside the network.

Image-only key/value cache. Table 5 compares two cache designs for current-centric sliding-window attention. The first variant caches both image tokens and pose-query tokens, while the second caches only image-conditioned states and re-instantiates pose queries for each current-centric window. Removing pose-query tokens from the cache improves ATE, RTE, and RRE on most scenes. This confirms that pose-query states are not generic reusable memory: they encode reasoning under a particular local gauge and can interfere with later predictions when propagated across windows. In contrast, image-conditioned states serve as reusable visual evidence and can be safely cached without explicit cache refresh. This ablation supports our second contribution: bounded GPU-memory streaming should cache image evidence while keeping pose reasoning transient and current-gauge-specific.

Active window size. We vary $W \in \{5, 10, 15\}$ using the ViT-Small model on vKITTI. Increasing W from 5 to 10 improves ATE/RRE/RTE from 3.23/0.12/0.15 to 2.64/0.09/0.10, while $W = 15$ provides only marginal gains (2.55/0.08/0.10) at higher cost (3.2 vs. 2.7 s/iter; 22.9 vs. 27.5 FPS). We therefore use $W = 10$ as a balance between accuracy and efficiency.

Runtime and memory scaling. Figure 5 compares Anchor3R with CUT3R [1], LongStream [6], FastVGGT [53], InfiniteVGGT [56], and SStream3R [4] under the same resolution and hardware. While several baselines exhibit increasing memory consumption or substantially higher runtime as the sequence grows, Anchor3R maintains nearly constant GPU memory up to 10K frames and scales approximately linearly in runtime. Notably, the model is trained on 48-frame sequences but directly generalizes to streams of up to 10K frames in this evaluation. The full model runs at 12.7 FPS with 11.9 GB peak GPU memory. Its runtime is comparable to CUT3R, while its GPU memory remains substantially lower than LongStream and InfiniteVGGT at long horizons. The accumulated motion graph and pointmaps are stored on CPU/disk and grow linearly with sequence length. Offline motion averaging remains lightweight, taking 11.88 s for 12K frames and 17.24 s for 18.8K frames.

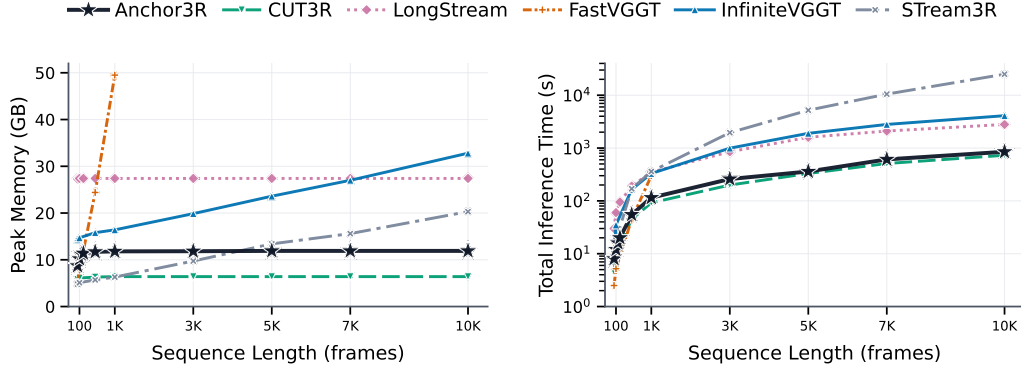


Figure 5: Runtime and GPU memory scaling under the same resolution and hardware. Anchor3R, trained on 48-frame sequences, maintains nearly constant GPU memory and approximately linear runtime up to 10K frames.

Controlled offline refinement. To isolate the effect of offline refinement, we apply the same loop-keyframe interval ($k = 5$) and motion-averaging protocol to LongStream and VGGT. LongStream-Off-5 improves over LongStream on 6/7 VBR sequences, while VGGT-Off-5 improves over VGGT-Long on all seven, confirming that loop-aware refinement benefits different prediction models. Under the same protocol, Anchor3R-Off-5 outperforms LongStream-Off-5 on all seven sequences and VGGT-Off-5 on 6/7 sequences, indicating that its offline gains cannot be attributed solely to the refinement backend. We further vary the keyframe interval $k \in \{5, 10, 15\}$ and observe consistent performance across most sequences, suggesting limited sensitivity to this choice.

5 Limitations

Anchor3R has several limitations. First, historical pointmaps are not updated after their initial prediction, which may limit global geometric consistency. Second, the current motion-averaging formulation assumes reasonably consistent relative scales and may degrade when scale estimates are unreliable. Third, although VBR, KITTI, and Waymo contain moving objects, we do not conduct dedicated evaluation on highly dynamic benchmarks, and robustness to severe scene dynamics remains to be studied. Finally, loop retrieval and global pose alignment remain separate from the learned reconstruction model; jointly learning long-range constraint discovery and global alignment is a promising direction toward a more fully end-to-end system. Our evaluation is also replay-based and does not include deployment on a physical robot.

6 Conclusion

We presented *Anchor3R*, a current-centric streaming 3D reconstruction framework for long-horizon visual mapping. Anchor3R treats feed-forward reconstruction as local measurement prediction: it predicts window-relative poses anchored at the current frame, caches reusable image-conditioned states while re-instantiating pose queries for each window, and integrates the resulting measurements through online pose updates and loop-aware motion averaging. This design avoids persistent global-gauge propagation inside the network, reduces stale pose-cache interference, and enables global drift redistribution through an explicit motion graph. Experiments on indoor, outdoor, driving, and RGB-D benchmarks show that Anchor3R improves long-horizon pose accuracy and dense map alignment over existing streaming feed-forward baselines, while controlled analyses validate the benefits of current-centric prediction, image-only caching, and graph-based pose recovery, as well as the scalability of Anchor3R to long visual streams.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Nos. U23A20386, 62572470, and U22B2055) and the Beijing Natural Science Foundation (No. L223003).

References

- [1] Q. Wang, Y. Zhang, A. Holynski, A. A. Efros, and A. Kanazawa. Continuous 3d perception model with persistent state. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 10510–10522, 2025.
- [2] X. Chen, Y. Chen, Y. Xiu, A. Geiger, and A. Chen. Ttt3r: 3d reconstruction as test-time training. *arXiv preprint arXiv:2509.26645*, 2025.
- [3] J. Dong, H. Li, S. Zhou, W. Hu, W. Xu, and Y. Wang. Memix: Writing less, remembering more for streaming 3d reconstruction. *arXiv preprint arXiv:2603.15330*, 2026.
- [4] Y. Lan, Y. Luo, F. Hong, S. Zhou, H. Chen, Z. Lyu, S. Yang, B. Dai, C. C. Loy, and X. Pan. Stream3r: Scalable sequential 3d reconstruction with causal transformer. *arXiv preprint arXiv:2508.10893*, 2025.
- [5] D. Zhuo, W. Zheng, J. Guo, Y. Wu, J. Zhou, and J. Lu. Streaming 4d visual geometry transformer. *arXiv preprint arXiv:2507.11539*, 2025.
- [6] C. Cheng, X. Chen, T. Xie, W. Yin, W. Ren, Q. Zhang, X. Guo, and H. Wang. Longstream: Long-sequence streaming autoregressive visual geometry. *arXiv preprint arXiv:2602.13172*, 2026.
- [7] Z. Li, J. Zhou, Y. Wang, H. Guo, W. Chang, Y. Zhou, H. Zhu, J. Chen, C. Shen, and T. He. Wint3r: Window-based streaming reconstruction with camera token pool. *arXiv preprint arXiv:2509.05296*, 2025.
- [8] C. Liu, J. Yang, Z. Li, Y. Deng, J. Guo, and L. Ballan. Mem3r: Streaming 3d reconstruction with hybrid memory via test-time training. *arXiv preprint arXiv:2604.07279*, 2026.
- [9] J. L. Schonberger and J.-M. Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4104–4113, 2016.
- [10] Y. Ding, J. Yang, V. Larsson, C. Olsson, and K. Åström. Revisiting the p3p problem. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4872–4880, 2023.
- [11] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.
- [12] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon. Bundle adjustment—a modern synthesis. In *International workshop on vision algorithms*, pages 298–372. Springer, 1999.
- [13] J. Ren, W. Liang, R. Yan, L. Mai, S. Liu, and X. Liu. Megba: A gpu-based distributed library for large-scale bundle adjustment. In *European Conference on Computer Vision*, pages 715–731. Springer, 2022.
- [14] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos. Orb-slam: A versatile and accurate monocular slam system. *IEEE transactions on robotics*, 31(5):1147–1163, 2015.
- [15] R. Mur-Artal and J. D. Tardós. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE transactions on robotics*, 33(5):1255–1262, 2017.
- [16] L. Pan, D. Baráth, M. Pollefeys, and J. L. Schönberger. Global structure-from-motion revisited. In *European Conference on Computer Vision*, pages 58–77. Springer, 2024.

- [17] S. Zhu, R. Zhang, L. Zhou, T. Shen, T. Fang, P. Tan, and L. Quan. Very large-scale global sfm by distributed motion averaging. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4568–4577, 2018.
- [18] A. Chatterjee and V. M. Govindu. Robust relative rotation averaging. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):958–972, 2017.
- [19] O. Ozyesil and A. Singer. Robust camera location estimation by convex programming. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2674–2683, 2015.
- [20] S. Wang, V. Leroy, Y. Cabon, B. Chidlovskii, and J. Revaud. Dust3r: Geometric 3d vision made easy. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 20697–20709, 2024.
- [21] J. Wang, M. Chen, N. Karaev, A. Vedaldi, C. Rupprecht, and D. Novotny. Vggt: Visual geometry grounded transformer. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 5294–5306, 2025.
- [22] Y. Wang, J. Zhou, H. Zhu, W. Chang, Y. Zhou, Z. Li, J. Chen, J. Pang, C. Shen, and T. He. π^3 : Permutation-equivariant visual geometry learning. *arXiv preprint arXiv:2507.13347*, 2025.
- [23] H. Lin, S. Chen, J. Liew, D. Y. Chen, Z. Li, G. Shi, J. Feng, and B. Kang. Depth anything 3: Recovering the visual space from any views. *arXiv preprint arXiv:2511.10647*, 2025.
- [24] N. Keetha, N. Müller, J. Schönberger, L. Porzi, Y. Zhang, T. Fischer, A. Knapitsch, D. Zauss, E. Weber, N. Antunes, et al. Mapanything: Universal feed-forward metric 3d reconstruction. *arXiv preprint arXiv:2509.13414*, 2025.
- [25] S. Elflein, R. Li, S. Agostinho, Z. Gojcic, L. Leal-Taixé, Q. Zhou, and A. Osep. Vgg-t³: Offline feed-forward 3d reconstruction at scale. *arXiv preprint arXiv:2602.23361*, 2026.
- [26] K. Deng, Z. Ti, J. Xu, J. Yang, and J. 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.
- [27] H. Jin, R. Wu, T. Zhang, R. Gao, J. T. Barron, N. Snavely, and A. Holynski. Zipmap: Linear-time stateful 3d reconstruction via test-time training. *arXiv preprint arXiv:2603.04385*, 2026.
- [28] J. Zhang, C. Herrmann, J. Hur, C. Sun, M.-H. Yang, F. Cole, T. Darrell, and D. Sun. Loger: Long-context geometric reconstruction with hybrid memory. *arXiv preprint arXiv:2603.03269*, 2026.
- [29] T. Xie, P. Yang, Y. Jin, Y. Cai, W. Yin, W. Ren, Q. Zhang, W. Hua, S. Peng, X. Guo, et al. Scal3r: Scalable test-time training for large-scale 3d reconstruction. *arXiv preprint arXiv:2604.08542*, 2026.
- [30] Y. Wu, W. Zheng, J. Zhou, and J. Lu. Point3r: Streaming 3d reconstruction with explicit spatial pointer memory. *arXiv preprint arXiv:2507.02863*, 2025.
- [31] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine learning*, 3(1):1–122, 2011.
- [32] H. Xia, Y. Fu, S. Liu, and X. Wang. Rgb-d objects in the wild: Scaling real-world 3d object learning from rgb-d videos, 2024.
- [33] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5828–5839, 2017.

- [34] M. Roberts, J. Ramapuram, A. Ranjan, A. Kumar, M. A. Bautista, N. Paczan, R. Webb, and J. M. Susskind. Hypersim: A photorealistic synthetic dataset for holistic indoor scene understanding. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10912–10922, 2021.
- [35] M. L. Antequera, P. Gargallo, M. Hofinger, S. R. Buló, Y. Kuang, and P. Kotschieder. Mapillary planet-scale depth dataset. In *European Conference on Computer Vision*, pages 589–604. Springer, 2020.
- [36] J. Straub, T. Whelan, L. Ma, Y. Chen, E. Wijmans, S. Green, J. J. Engel, R. Mur-Artal, C. Ren, S. Verma, et al. The replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019.
- [37] E. Arnold, J. Wynn, S. Vicente, G. Garcia-Hernando, A. Monszpart, V. Prisacariu, D. Turmukhambetov, and E. Brachmann. Map-free visual relocalization: Metric pose relative to a single image. In *European Conference on Computer Vision*, pages 690–708. Springer, 2022.
- [38] W. Wang, D. Zhu, X. Wang, Y. Hu, Y. Qiu, C. Wang, Y. Hu, A. Kapoor, and S. Scherer. Tartanair: A dataset to push the limits of visual slam. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4909–4916. IEEE, 2020.
- [39] P.-H. Huang, K. Matzen, J. Kopf, N. Ahuja, and J.-B. Huang. Deepmvs: Learning multi-view stereopsis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2821–2830, 2018.
- [40] Y. Cabon, N. Murray, and M. Humenberger. Virtual kitti 2, 2020.
- [41] X. Pan, N. Charron, Y. Yang, S. Peters, T. Whelan, C. Kong, O. Parkhi, R. Newcombe, and Y. C. Ren. Aria digital twin: A new benchmark dataset for egocentric 3d machine perception. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 20133–20143, 2023.
- [42] L. Mehl, J. Schmalfluss, A. Jahedi, Y. Nalivayko, and A. Bruhn. Spring: A high-resolution high-detail dataset and benchmark for scene flow, optical flow and stereo. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4981–4991, 2023.
- [43] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2446–2454, 2020.
- [44] Y. Yao, Z. Luo, S. Li, J. Zhang, Y. Ren, L. Zhou, T. Fang, and L. Quan. Blendedmvs: A large-scale dataset for generalized multi-view stereo networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1790–1799, 2020.
- [45] J. Reizenstein, R. Shapovalov, P. Henzler, L. Sbordone, P. Labatut, and D. Novotny. Common objects in 3d: Large-scale learning and evaluation of real-life 3d category reconstruction. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10901–10911, 2021.
- [46] Z. Li and N. Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2041–2050, 2018.
- [47] L. Ling, Y. Sheng, Z. Tu, W. Zhao, C. Xin, K. Wan, L. Yu, Q. Guo, Z. Yu, Y. Lu, et al. D13dv-10k: A large-scale scene dataset for deep learning-based 3d vision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22160–22169, 2024.

- [48] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.
- [49] L. Brizi, E. Giacomini, L. Di Giammarino, S. Ferrari, O. Salem, L. De Rebotti, and G. Grisetti. Vbr: A vision benchmark in rome. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 15868–15874. IEEE, 2024.
- [50] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers. A benchmark for the evaluation of rgb-d slam systems. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pages 573–580. IEEE, 2012.
- [51] Y. Tao, M. Á. Muñoz-Bañón, L. Zhang, J. Wang, L. F. T. Fu, and M. Fallon. The oxford spires dataset: Benchmarking large-scale lidar-visual localisation, reconstruction and radiance field methods. *International Journal of Robotics Research*, 2025.
- [52] J. Shotton, B. Glocker, C. Zach, S. Izadi, A. Criminisi, and A. Fitzgibbon. Scene coordinate regression forests for camera relocalization in rgb-d images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2930–2937, 2013.
- [53] Y. Shen, Z. Zhang, Y. Qu, X. Zheng, J. Ji, S. Zhang, and L. Cao. Fastvggt: Training-free acceleration of visual geometry transformer. *arXiv preprint arXiv:2509.02560*, 2025.
- [54] R. Murai, E. Dexheimer, and A. J. Davison. Mast3r-slam: Real-time dense slam with 3d reconstruction priors. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 16695–16705, 2025.
- [55] D. Maggio, H. Lim, and L. Carlone. Vggt-slam: Dense rgb slam optimized on the sl (4) manifold. *arXiv preprint arXiv:2505.12549*, 2025.
- [56] S. Yuan, Y. Yang, X. Yang, X. Zhang, Z. Zhao, L. Zhang, and Z. Zhang. Infinitevggt: Visual geometry grounded transformer for endless streams. *arXiv preprint arXiv:2601.02281*, 2026.
- [57] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic. Netvlad: Cnn architecture for weakly supervised place recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5297–5307, 2016.

Supplementary Material

A Online Motion Averaging

During online inference, Anchor3R does not compose poses along a single temporal chain. Instead, the current pose is estimated from all available window-relative measurements to previously recovered frames. At time t , for each historical frame $i \in \mathcal{W}_t \setminus \{t\}$, the network predicts a relative rotation $\hat{\mathbf{R}}_{i \leftarrow t}$ satisfying

$$\hat{\mathbf{R}}_{i \leftarrow t} \approx \mathbf{R}_i \mathbf{R}_t^\top. \quad (3)$$

Given the previously estimated global rotation $\hat{\mathbf{R}}_i$, this edge induces a candidate current rotation

$$\tilde{\mathbf{R}}_t^{(i)} = \hat{\mathbf{R}}_{i \leftarrow t}^\top \hat{\mathbf{R}}_i. \quad (4)$$

Rather than selecting one candidate or averaging rotations in Euclidean space, we compute a robust Lie-algebra median over the candidate rotations $\{\tilde{\mathbf{R}}_t^{(i)}\}_{i \in \mathcal{W}_t \setminus \{t\}}$. This makes the online update robust to occasional inaccurate relative-pose predictions caused by weak overlap, motion blur, or transient matching ambiguity.

Specifically, we initialize multiple rotation hypotheses by randomly sampling from the candidate set. For each hypothesis $\mathbf{R}^{(m)}$, we compute residual rotations $\tilde{\mathbf{R}}_t^{(i)} (\mathbf{R}^{(m)})^\top$ and map them to the Lie algebra:

$$\mathbf{r}_i^{(m)} = \text{Log}\left(\tilde{\mathbf{R}}_t^{(i)} (\mathbf{R}^{(m)})^\top\right) \in \mathbb{R}^3. \quad (5)$$

The hypothesis is updated by the coordinate-wise median residual:

$$\Delta\omega^{(m)} = \text{median}_i\{\mathbf{r}_i^{(m)}\}, \quad \mathbf{R}^{(m)} \leftarrow \text{Exp}(\Delta\omega^{(m)})\mathbf{R}^{(m)}. \quad (6)$$

The iteration stops when $\|\Delta\omega^{(m)}\| < \epsilon$. Among all converged hypotheses, we select the one with the smallest median angular residual. We use stable implementations of the SO(3) logarithm and exponential maps, including small-angle handling, and use multiple random initializations to reduce sensitivity to the initial hypothesis. If no hypothesis converges, we fall back to the first candidate rotation. We summarize this robust online rotation update in Algorithm 1.

The current camera center is estimated in the same robust manner but does not require iterative optimization. Each historical frame provides a candidate current center by adding its recovered position to the predicted relative translation:

$$\tilde{\mathbf{c}}_t^{(i)} = \hat{\mathbf{c}}_i + \hat{\mathbf{v}}_{i,t}, \quad \hat{\mathbf{v}}_{i,t} \approx \mathbf{c}_t - \mathbf{c}_i. \quad (7)$$

We then take the coordinate-wise median over all candidates,

$$\hat{\mathbf{c}}_t = \text{median}_{i \in \mathcal{W}_t \setminus \{t\}}\{\tilde{\mathbf{c}}_t^{(i)}\}. \quad (8)$$

This yields a simple and robust online translation update, consistent with the median-based rotation estimate. Together, the rotation and translation medians allow Anchor3R to integrate multiple current-centric measurements at each step without relying on a single fragile temporal edge.

B Offline Motion Averaging

After streaming inference, Anchor3R refines the complete trajectory by optimizing over the accumulated window-relative measurements. Unlike online pose averaging, which estimates each new pose from only the current active window, offline motion averaging uses all local edges produced along the sequence, including additional long-range edges from loop-closure reinsertion. This step is enabled by our current-centric task definition: each prediction is a reusable relative-pose constraint rather than an absolute pose tied to a historical coordinate system.

Algorithm 1 Online Lie-Algebra Median Rotation Averaging

Require: Candidate rotations $\{\tilde{\mathbf{R}}_j\}_{j=1}^M$, number of initializations N_{init} , maximum iterations K , threshold ϵ

Ensure: Robust current rotation $\hat{\mathbf{R}}_t$

- 1: Randomly initialize $\{\mathbf{R}^{(m)}\}_{m=1}^{N_{\text{init}}}$ from $\{\tilde{\mathbf{R}}_j\}_{j=1}^M$
- 2: Set scores $e^{(m)} \leftarrow +\infty$ and convergence flags $c^{(m)} \leftarrow \text{false}$
- 3: **for** $k = 1$ to K **do**
- 4: **for** $m = 1$ to N_{init} **do**
- 5: **if** $c^{(m)} = \text{true}$ **then**
- 6: **continue**
- 7: **end if**
- 8: **for** $j = 1$ to M **do**
- 9: $\mathbf{r}_j^{(m)} \leftarrow \text{Log}(\tilde{\mathbf{R}}_j(\mathbf{R}^{(m)})^\top)$
- 10: $\theta_j^{(m)} \leftarrow \|\mathbf{r}_j^{(m)}\|_2$
- 11: **end for**
- 12: $\Delta\omega^{(m)} \leftarrow \text{median}_j\{\mathbf{r}_j^{(m)}\}$
- 13: **if** $\|\Delta\omega^{(m)}\|_2 < \epsilon$ **then**
- 14: $e^{(m)} \leftarrow \text{median}_j\{\theta_j^{(m)}\}$
- 15: $c^{(m)} \leftarrow \text{true}$
- 16: **else**
- 17: $\mathbf{R}^{(m)} \leftarrow \text{Exp}(\Delta\omega^{(m)})\mathbf{R}^{(m)}$
- 18: **end if**
- 19: **end for**
- 20: **if** all hypotheses have converged **then**
- 21: **break**
- 22: **end if**
- 23: **end for**
- 24: **if** at least one hypothesis converged **then**
- 25: $m^* \leftarrow \arg \min_m e^{(m)}$
- 26: $\hat{\mathbf{R}}_t \leftarrow \mathbf{R}^{(m^*)}$
- 27: **else**
- 28: $\hat{\mathbf{R}}_t \leftarrow \tilde{\mathbf{R}}_1$
- 29: **end if**
- 30: **return** $\hat{\mathbf{R}}_t$

Rotation averaging. Let F be the number of frames and let $S = F - W + 1$ be the number of sliding windows. For each window $s \in \{1, \dots, S\}$ and each source frame $j \in \{0, \dots, W - 2\}$, Anchor3R predicts a relative rotation $\hat{\mathbf{R}}_{s+j \leftarrow s+W-1}$, where the last frame of the window is the current anchor. We initialize the global rotations with the online estimates and remove the gauge by setting the frame at index $W - 1$ as the reference:

$$\mathbf{R}_i^{(0)} = \hat{\mathbf{R}}_i^{\text{online}} \left(\hat{\mathbf{R}}_{W-1}^{\text{online}} \right)^\top. \quad (9)$$

For each relative rotation edge, the residual is computed in the Lie algebra as

$$\mathbf{r}_{s,j} = \text{Log} \left((\mathbf{R}_{s+j})^\top \hat{\mathbf{R}}_{s+j \leftarrow s+W-1} \mathbf{R}_{s+W-1} \right) \in \mathbb{R}^3. \quad (10)$$

The gauge-fixing residual is $\mathbf{r}_{\text{fix}} = \text{Log}(\mathbf{R}_{W-1}^\top)$. At each IRLS iteration, we solve for incremental axis-angle updates $\{\Delta\omega_i\}_{i=1}^F$ with a sparse linearized system. For an edge $(s + j, s + W - 1)$, the corresponding linearized constraint is

$$\Delta\omega_{s+j} - \Delta\omega_{s+W-1} \approx \mathbf{r}_{s,j}. \quad (11)$$

We apply a robust Geman–McClure-style weight to each residual,

$$w_{s,j} = \frac{\sigma^2}{(\sigma^2 + \|\mathbf{r}_{s,j}\|_2^2)^2}, \quad \sigma = 5^\circ, \quad (12)$$

Algorithm 2 Offline Motion Averaging

Require: Window-relative poses $\{\hat{\mathbf{R}}_{s+j \leftarrow s+W-1}, \hat{\mathbf{t}}_{s+j \leftarrow s+W-1}\}$, online rotations $\{\hat{\mathbf{R}}_i^{\text{online}}\}$, window size W

Ensure: Refined global rotations $\{\mathbf{R}_i\}$ and camera centers $\{\mathbf{c}_i\}$

- 1: Initialize $\mathbf{R}_i \leftarrow \hat{\mathbf{R}}_i^{\text{online}} (\hat{\mathbf{R}}_{W-1}^{\text{online}})^\top$
 - 2: Build sparse rotation incidence matrix $\mathbf{A}$ with gauge fixed at frame $W-1$
 - 3: **for** $k = 1$ to K_{rot} **do**
 - 4: Compute Lie residuals $\mathbf{r}_{s,j} \leftarrow \text{Log}((\mathbf{R}_{s+j})^\top \hat{\mathbf{R}}_{s+j \leftarrow s+W-1} \mathbf{R}_{s+W-1})$
 - 5: Compute robust weights $w_{s,j} \leftarrow \sigma^2 / (\sigma^2 + \|\mathbf{r}_{s,j}\|_2^2)$
 - 6: Solve $(\mathbf{A}^\top \mathbf{W} \mathbf{A}) \Delta \boldsymbol{\omega} = \mathbf{A}^\top \mathbf{W} \mathbf{r}$
 - 7: Update $\mathbf{R}_i \leftarrow \mathbf{R}_i \text{Exp}(\Delta \boldsymbol{\omega}_i)$ for all frames
 - 8: **if** $\frac{1}{F} \sum_i \|\Delta \boldsymbol{\omega}_i\|_2 < \epsilon_{\text{rot}}$ **then**
 - 9: **break**
 - 10: **end if**
 - 11: **end for**
 - 12: Rotate relative translations into the global frame: $\hat{\mathbf{v}}_{s,j} \leftarrow \mathbf{R}_{s+j}^\top \hat{\mathbf{t}}_{s+j \leftarrow s+W-1}$
 - 13: Build sparse LAD system over camera centers $\{\mathbf{c}_i\}$ with fixed scale, enforcing $\mathbf{c}_{s+j} - \mathbf{c}_{s+W-1} + \hat{\mathbf{v}}_{s,j} \approx \mathbf{0}$
 - 14: Solve $\min_{\mathbf{c}} \|\mathbf{A}_t \mathbf{c} - \mathbf{b}_t\|_1$ with ADMM and sparse Cholesky
 - 15: **return** $\{\mathbf{R}_i\}, \{\mathbf{c}_i\}$
-

and solve the weighted normal equation

$$(\mathbf{A}^\top \mathbf{W} \mathbf{A}) \Delta \boldsymbol{\omega} = \mathbf{A}^\top \mathbf{W} \mathbf{r}. \quad (13)$$

The rotations are then updated by right multiplication:

$$\mathbf{R}_i \leftarrow \mathbf{R}_i \text{Exp}(\Delta \boldsymbol{\omega}_i). \quad (14)$$

We repeat this process until the average update magnitude is below a small threshold or the maximum number of iterations is reached.

Position averaging. After rotation averaging, we recover camera centers from all window-relative translation measurements. For each edge $(s+j, s+W-1)$, we first rotate the predicted relative translation into the global frame:

$$\hat{\mathbf{v}}_{s,j} = \mathbf{R}_{s+j}^\top \hat{\mathbf{t}}_{s+j \leftarrow s+W-1} \approx \mathbf{c}_{s+W-1} - \mathbf{c}_{s+j}. \quad (15)$$

We then estimate the global camera centers by directly enforcing the relative translation constraints:

$$\mathbf{c}_{s+j} - \mathbf{c}_{s+W-1} + \hat{\mathbf{v}}_{s,j} = \mathbf{0}, \quad j = 0, \dots, W-2. \quad (16)$$

The trajectory gauge is fixed by setting one reference center, e.g., $\mathbf{c}_{W-1} = \mathbf{0}$. This gives a sparse L1 translation averaging problem:

$$\min_{\{\mathbf{c}_i\}} \sum_{s=1}^S \sum_{j=0}^{W-2} \|\mathbf{c}_{s+j} - \mathbf{c}_{s+W-1} + \hat{\mathbf{v}}_{s,j}\|_1, \quad \text{s.t.} \quad \mathbf{c}_{W-1} = \mathbf{0}. \quad (17)$$

Here the L1 objective improves robustness to inaccurate relative translations from weak-overlap or ambiguous windows. Unlike the scale-variable formulation, this direct form does not allow each window to absorb errors with an independent scale, and therefore better preserves the scale consistency learned between relative poses and local pointmaps. In implementation, we assemble the constraints into a sparse linear system and solve the LAD problem with ADMM, where the x -step is accelerated by sparse Cholesky factorization. The resulting camera centers, together with the averaged rotations, define the final global trajectory used to align local pointmaps into a coherent reconstruction.

Table 6: **Additional ablation on KITTI.** We compare Anchor3R-Online and Anchor3R-Offline without loop closure on all 11 KITTI Odometry sequences. Offline w/o LC optimizes only the dense window-relative measurements accumulated during streaming inference, without loop-closure reinsertion. Best RRE and RTE values between online and offline variants are highlighted in bold.

Sequence	#Frames	Pose-Online		Pose-Offline w/o LC	
		RRE ($^{\circ}$) $\downarrow$	RTE (m) $\downarrow$	RRE ($^{\circ}$) $\downarrow$	RTE (m) $\downarrow$
00	4541	0.200	0.084	0.191	0.072
01	1101	0.220	0.334	0.203	0.305
02	4661	0.198	0.234	0.188	0.234
03	801	0.152	0.072	0.137	0.063
04	271	0.129	0.137	0.120	0.116
05	2761	0.156	0.105	0.149	0.096
06	1101	0.141	0.105	0.134	0.087
07	1101	0.174	0.082	0.166	0.075
08	4071	0.182	0.148	0.168	0.144
09	1591	0.212	0.231	0.202	0.229
10	1201	0.180	0.081	0.169	0.070
Avg.	—	0.177	0.147	0.166	0.136

Table 7: **Controlled offline refinement on VBR.** We apply the same loop-keyframe retrieval and motion-averaging protocol to LongStream, VGGT, and Anchor3R. Off- k denotes a loop-keyframe interval of k . Lower ATE is better.

Method	campus_train0	campus_train1	ciampino_train1	colosseo_train0	diag_train0	pincio_train0	spagna_train0	Avg.
LongStream	100.57	105.55	131.78	72.52	32.35	43.47	59.31	77.93
LongStream-Off-5	87.61	99.86	98.64	45.99	8.46	65.84	39.77	63.74
VGGT-Long	118.59	98.21	172.13	39.56	30.80	53.44	50.27	80.43
VGGT-Off-5	12.20	9.85	28.30	20.21	12.84	19.77	15.56	16.96
Anchor3R-Off-5	5.13	3.52	78.64	17.25	5.35	15.56	11.76	19.60
Anchor3R-Off-10	5.02	2.91	92.51	19.15	5.21	16.51	11.27	21.80
Anchor3R-Off-15	5.15	3.62	100.87	21.65	5.04	18.77	12.11	23.89

C Additional Ablation on Online and Offline Motion Averaging

We further ablate the effect of offline motion averaging without loop closure on KITTI. This setting isolates whether the dense window-relative measurements accumulated during online streaming can improve pose consistency by themselves, without introducing additional long-range loop constraints.

As shown in Table 6, offline motion averaging consistently improves pose accuracy even without loop closure. Compared with the online trajectory, Offline w/o LC reduces the average RRE from 0.177° to 0.166° and the average RTE from 0.147 m to 0.136 m . This confirms that the current-centric dense relative-pose predictions provide useful redundant constraints for global pose refinement, rather than serving only as one-step online estimates.

D Controlled Offline Refinement on VBR

Applying the same offline refinement to other prediction models also improves their trajectories: LongStream-Off-5 improves over LongStream on 6 of 7 sequences, while VGGT-Off-5 improves over VGGT-Long on all seven. Under the same protocol, Anchor3R-Off-5 outperforms LongStream-Off-5 on all seven sequences and VGGT-Off-5 on 6 of 7 sequences, indicating that the gains cannot be attributed solely to the shared refinement backend.

We further vary the loop-keyframe interval for Anchor3R. Performance remains broadly consistent on six of seven sequences for $k \in \{5, 10, 15\}$, while *ciampino_train1* becomes more challenging as the interval increases. This suggests that Anchor3R is generally insensitive to moderate changes in loop-keyframe density, although difficult trajectories can benefit from denser long-range constraints.

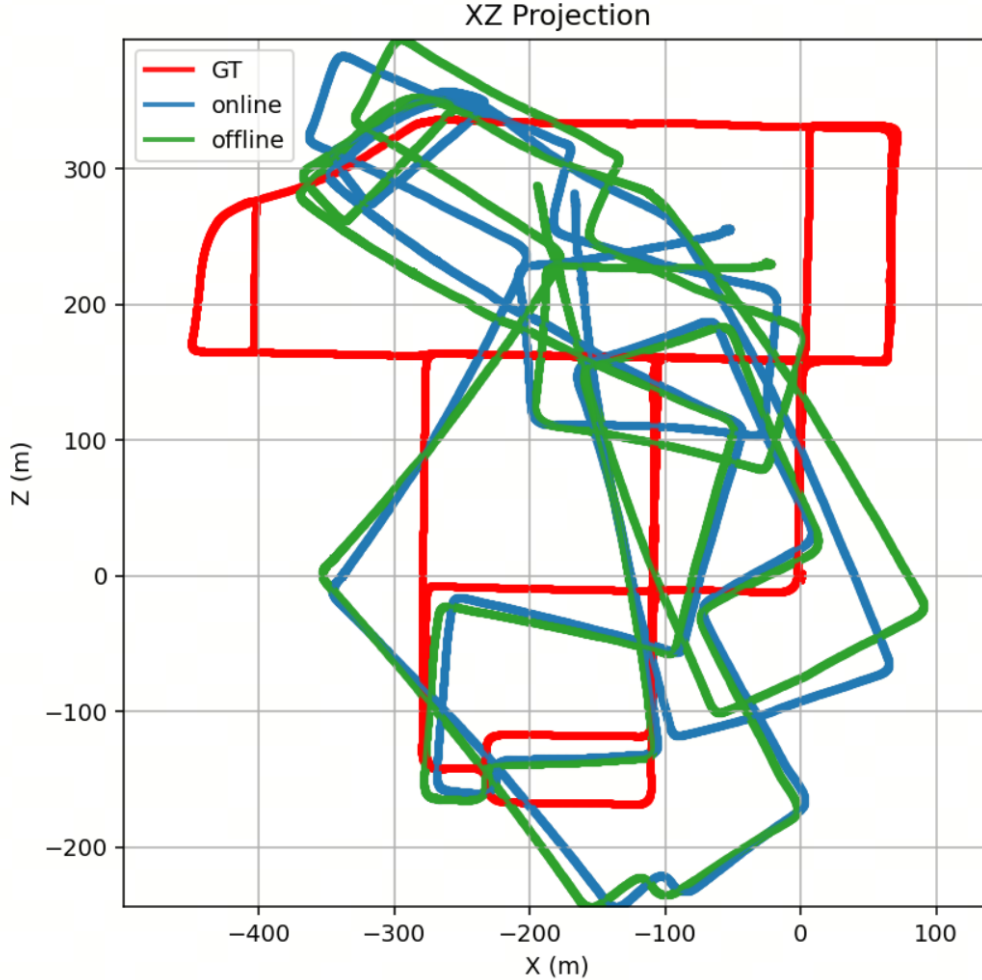


Figure 6: **Failure case on *ciampino_train1*.** We show the online trajectory and offline refinement without loop-closure constraints. Errors are most evident around frequent turns, where orientation errors affect subsequent translation directions and accumulate over the long trajectory.

E Failure Case Analysis

Figure 6 shows a representative failure case on *ciampino_train1*. Both the online result and offline refinement without loop closure exhibit noticeable drift around regions with frequent turns. Small orientation errors alter the directions of subsequent translations and can therefore accumulate over a long trajectory. Offline motion averaging redistributes locally redundant measurements, but without additional long-range constraints it cannot fully eliminate this accumulated drift. This example highlights the remaining challenge of maintaining robust orientation estimates under repeated large rotations and motivates stronger long-range constraint discovery in future work.

F Evaluation Dataset Details

All evaluations are conducted on complete sequences without frame subsampling. We summarize the dataset-specific settings below.

VBR. Following the protocol of LoGeR [28], we use all 7 full-length sequences, which contain 8,815–18,846 frames and cover trajectories up to 5.2 km.

KITTI. For KITTI Odometry, we evaluate the full camera-02 image streams from all 11 sequences, i.e., sequences 00–10.

Waymo Open. We evaluate on 9 held-out segments that are excluded from our training set: 163453191 (198 frames, 160 m), 183829460 (199 frames, 42 m), 315615587 (199 frames, 165 m), 346181117 (199 frames, 351 m), 371159869 (196 frames, 273 m), 405841035 (199 frames, 86 m), 460417311 (198 frames, 266 m), 520018670 (199 frames, 135 m), and 610454533 (198 frames, 63 m). Although Waymo is included in the training data, these held-out segments are used to test generalization to unseen driving scenes.

Oxford Spires. We use the front-camera images and evaluate on all 12 subsets.

7Scenes. For each scene in 7Scenes, including Chess, Fire, Heads, Office, Pumpkin, RedKitchen, and Stairs, we evaluate on sequence 01.

TUM RGB-D. We follow the standard full-sequence evaluation protocol.