CLoSeR: Closing the Loop for Long-Context Streaming Reconstruction

Moyang Li[1*] **Zihan Zhu**[1*] **Wei Zhang**[2] **Marc Pollefeys**[1,3] **Daniel Barath**[1]
[1]ETH Zurich [2]University of Stuttgart [3]Microsoft

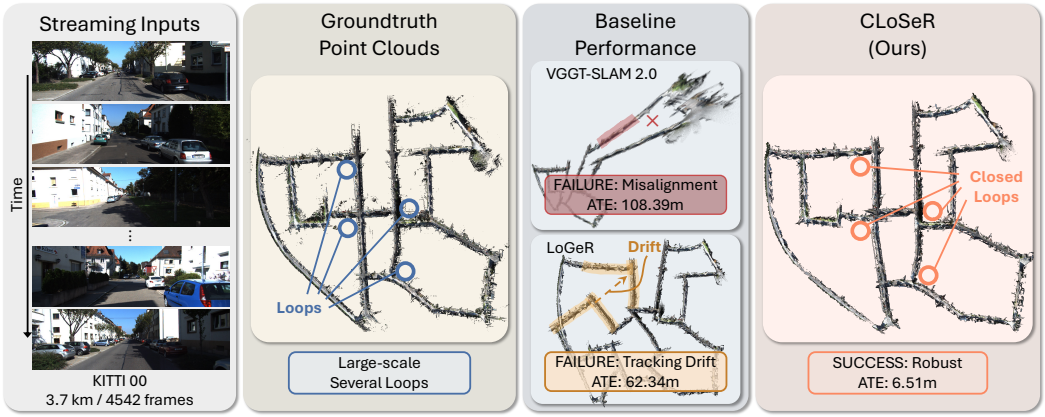

Figure 1: **CLoSeR.** For very long sequences, the submap-based approach VGGT-SLAM 2.0 exhibits erroneous point cloud alignment, while the feedforward method LoGeR suffers from tracking drift due to the absence of explicit constraints when revisiting regions. In contrast, our approach consistently produces accurate reconstructions at kilometer scale.

Abstract

Feedforward foundation models have recently shown remarkable 3D reconstruction capabilities. However, existing models exhibit large tracking drift in long-context streaming reconstruction due to error accumulation. In this paper, we revisit loop closure with streaming reconstruction foundation models to enable accurate, drift-free, kilometer-scale reconstruction. Specifically, our method detects loop candidates through global descriptor retrieval, and constructs loop-conditioned windows to estimate the relative poses between looped frames. Given the observation that our adopted streaming reconstruction backbone produces a globally consistent scale, we optimize all frame poses on the SE(3) manifold with sequential and loop closure constraints, avoiding the pose graph optimization on the Sim(3) or higher-dimensional SL(4) manifolds employed in prior works. Extensive experiments show that our method reduces drift and produces consistent geometry on kilometer-scale sequences, significantly outperforming the state of the art. Code is available at https://github.com/MoyangLi00/CLoSeR.git.

1 Introduction

Accurate geometry and pose estimation from a sequence of images are fundamental yet challenging tasks in 3D vision, with broad applications in robotics [Liu et al., 2024, Arm et al., 2023, Zhou et al., 2022] and autonomous driving [Badue et al., 2021, Geiger et al., 2013]. Traditional approaches

*Authors contributed equally to this work. Author order is interchangeable.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

address this problem by leveraging multi-view geometric constraints, ranging from feature-based pipelines [Mur-Artal et al., 2015, Mur-Artal and Tardós, 2017, Campos et al., 2021] that triangulate sparse keypoints across frames, to direct methods [Engel et al., 2017, 2014] that minimize photometric residuals over image intensities. To achieve more robust estimation under challenging conditions, additional sensors such as inertial measurement units (IMU) [Qin et al., 2018, Leutenegger, 2022] and LiDAR [Zhang et al., 2014, Shan et al., 2020, Xu and Zhang, 2021] are commonly fused into the pipeline. Moreover, some systems have extensively studied loop closure detection [Gálvez-López and Tardos, 2012, Arandjelovic et al., 2016] and pose graph optimization [Kümmerle et al., 2011, Kaess et al., 2012], to correct the tracking drift over long trajectories and large-scale scenes.

Recently, feedforward foundation models have shifted the paradigm in 3D vision. Early works [Wang et al., 2024b, Leroy et al., 2024, Chen et al., 2025a] estimate dense two-view geometry and camera poses with transformers, and rely on pairwise alignment to handle multiple frames. VGGT [Wang et al., 2025a] extends this paradigm to multi-frame inputs, jointly predicting camera poses, depth maps, dense point clouds, and camera intrinsics in a single forward pass. However, the $O(n^2)$ computational complexity of self-attention layers and its non-causal architecture prevent VGGT from scaling to long sequences or supporting streaming inference.

Several approaches extend feedforward backbones to long-sequence and sequential reconstruction. Some streaming approaches adopt online updates of global memory representation [Wang et al., 2025b, Chen et al., 2025b] or Test-Time Training (TTT) layers [Zhuo et al., 2025, Elflein et al., 2026, Jin et al., 2026, Zhang et al., 2026] to propagate information across windows to enable long-context reconstruction. However, they still suffer from tracking drift over long trajectories due to the lack of explicit constraints imposed on revisited regions. Submap-based approaches [Maggio et al., 2025, Maggio and Carlone, 2026, Deng et al., 2025] instead partition the sequence into chunks, independently infer the geometry and camera poses of each submap, and align them through point clouds registration. These approaches further incorporate loop closure to mitigate drift. However, since each submap is produced by an independent forward pass of VGGT, scale inconsistency inevitably arises across submaps. As a consequence, the resulting pose graph has to be optimized on the Sim(3) [Deng et al., 2025] or even higher-dimensional SL(4) [Maggio et al., 2025, Maggio and Carlone, 2026] manifold to absorb scale or projective ambiguity. However, the relative poses and scales recovered through point cloud registration remain noisy and unreliable, as shown in Fig. 1, which substantially limits their performance in real-world complex scenarios.

This raises a natural question: *can loop closure be made simpler and more reliable when built upon a streaming backbone whose global scale is already consistent?* In this paper, we present **CLoSeR**, a streaming reconstruction system that augments a prior streaming backbone LoGeR [Zhang et al., 2026] with loop closure for accurate kilometer-scale reconstruction from uncalibrated monocular videos. Our approach exploits two key properties of LoGeR: (a) its pose predictions across windows share a globally consistent scale, rendering pose graph optimization on the SE(3) manifold sufficient and avoiding the Sim(3) or SL(4) manifolds used in prior submap-based methods; (b) its window-wise inference does not require temporal contiguity of input frames, allowing us to append a loop-conditioned window comprising the current and revisited frames after the regular streaming windows, and to directly recover their relative poses in the global coordinate frame.

In summary, our main contributions are as follows:

- We present a system that integrates loop closure into streaming feedforward reconstruction models, enabling accurate kilometer-scale reconstruction from monocular videos.
- We propose a simple yet effective loop-conditioned window mechanism that exploits the streaming nature of the backbone to recover relative poses between looped frames in a globally consistent coordinate frame.
- We show that, owing to the globally consistent scale, pose graph optimization on the SE(3) manifold is sufficient for accurate large-scale reconstruction, obviating the Sim(3) or SL(4) optimization and the point-cloud alignment adopted by prior submap-based methods.

2 Related Works

Structure-from-Motion (SfM) aims to estimate sparse point clouds and camera poses from a collection of images. Traditional SfM pipelines [Snavely et al., 2008, Schönberger and Frahm, 2016, Pan et al., 2024] typically decompose this problem into several stages: keypoint detection and correspondence matching, camera pose estimation, and joint refinement of structure and poses. To

improve the robustness and accuracy of SfM, recent works incorporate deep learning at different stages, including deep feature extraction [DeTone et al., 2018, Dusmanu et al., 2019, Yi et al., 2016] and learning-based feature matching [Sarlin et al., 2020, Edstedt et al., 2024, Pautrat et al., 2023, Lindenberger et al., 2023]. Beyond this, another line of work explores end-to-end differentiable SfM [Teed and Deng, 2018, Wei et al., 2020, Brachmann et al., 2024, Wang et al., 2024a], which jointly optimizes geometry and camera poses within a unified learning framework.

Simultaneous Localization and Mapping (SLAM) aims to estimate camera poses and reconstruct the scene online from sequential observations. Traditional approaches are commonly categorized into feature-based methods [Mur-Artal et al., 2015, Mur-Artal and Tardós, 2017, Campos et al., 2021] and direct methods [Engel et al., 2017, 2014, Newcombe et al., 2011]. Feature-based methods track sparse keypoints across frames and triangulate them into a sparse map, while direct methods minimize photometric residuals over image intensities to jointly recover poses and geometry. To improve robustness under challenging conditions like motion blur, low texture and large-scale scenes, some pipelines introduce additional sensors (IMU [Qin et al., 2018, Leutenegger, 2022, Zhu et al., 2026] and LiDAR [Zhang et al., 2014, Shan et al., 2020, Xu and Zhang, 2021]). To mitigate drift over long trajectories, many systems further incorporate loop closure [Mur-Artal et al., 2015, Mur-Artal and Tardós, 2017, Campos et al., 2021, Engel et al., 2014, Leutenegger, 2022]. More recently, learning-based visual SLAM and VO systems [Teed and Deng, 2021, Teed et al., 2023, Lipson et al., 2024] have demonstrated superior accuracy and robustness over their classical counterparts. In parallel, another line of work integrates neural scene representations such as NeRF [Zhu et al., 2022, 2024] and 3D Gaussian Splatting [Yan et al., 2024, Keetha et al., 2024, Matsuki et al., 2024] into SLAM, enabling photorealistic dense mapping. Despite maintaining consistency over long sequences, these carefully designed SLAM systems typically rely on iterative optimization and sophisticated pipelines. In contrast, recent feedforward reconstruction models jointly predict scene geometry and camera poses in a single forward pass, eliminating the need for iterative optimization.

Feedforward 3D Reconstruction Models predict dense 3D geometry directly from images without requiring known camera parameters. DUSt3R [Wang et al., 2024b] and MASt3R [Leroy et al., 2024] predict dense point clouds from an image pair within a common coordinate frame. MonST3R [Zhang et al., 2024] and Easi3R [Chen et al., 2025a] further extend DUSt3R to dynamic environments. VGGT [Wang et al., 2025a] and π^3 [Wang et al., 2025c] support arbitrary numbers of input frames and achieve more accurate predictions. However, their quadratic attention limits scalability to very long sequences and prohibits online streaming inference. To address this limitation, VGGT-SLAM [Maggio et al., 2025, Maggio and Carlone, 2026] and VGGT-Long [Deng et al., 2025] extend VGGT to long sequences through submap-based alignment. In contrast, CUT3R [Wang et al., 2025b] and TTT3R [Chen et al., 2025b] handle long sequences by continuously updating a global state representation in an online manner. Several other streaming reconstruction works [Elflein et al., 2026, Jin et al., 2026, Zhang et al., 2026] instead leverage Test-Time Training (TTT) layers [Sun et al., 2024, Zhang et al., 2025] to propagate information across windows with linear complexity. Despite the rapid progress of feedforward reconstruction, existing approaches still struggle with drift accumulation in long-context streaming reconstruction. In contrast, our method incorporates effective loop closure within the feedforward reconstruction model, enabling accurate, drift-free streaming reconstruction at the kilometer scale.

3 Method

Our proposed method, **CLoSeR**, enables accurate kilometer-scale streaming reconstruction from monocular sequences by augmenting a feedforward backbone [Zhang et al., 2026] with a loop closure module. In Sec. 3.1, we introduce the preliminaries of the streaming reconstruction backbone that our framework builds upon. In Sec. 3.2, we present loop detection and relative pose estimation between looped frames. Finally, in Sec. 3.3, we describe the global pose graph optimization over all frames with sequential and loop closure constraints. The overview of CLoSeR is shown in Fig. 2.

3.1 Preliminaries

Given a long image sequence $\mathcal{I} = \{\mathbf{I}_1, \dots, \mathbf{I}_N\}$ as inputs, LoGeR [Zhang et al., 2026] partitions it into K overlapping chunks of size N_{win} with stride $s = N_{\text{win}} - N_{\text{overlap}}$. The k-th chunk ($k = 1, \dots, K$) is defined as

$$\mathcal{W}_k = \{\, \mathbf{I}_t \mid (k-1)s + 1 \le t \le (k-1)s + N_{\text{win}} \,\}. \tag{1}$$

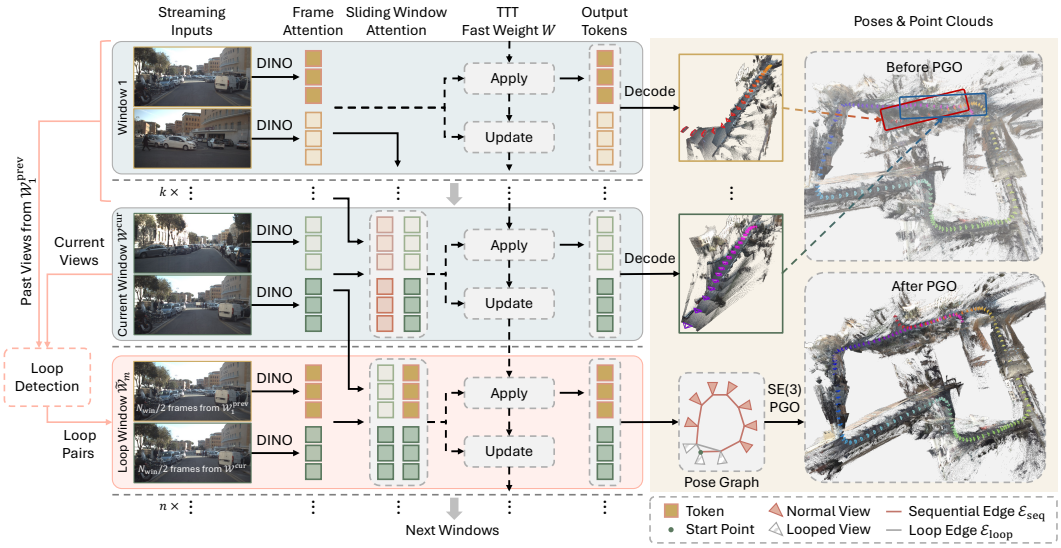

Figure 2: **System overview.** Given a streaming monocular sequence, our method processes it in a sliding-window manner. For each window, we first tokenize the frames with a DINO-based encoder and then apply a stack of attention layers, comprising frame attention, sliding-window attention, and TTT layers, to maintain global consistency across the sequence. Upon the arrival of each new window, we additionally extract SALAD descriptors to identify potential loop closures against prior windows. When such candidates are detected, we construct a loop-conditioned window by pairing $N_{\text{win}}/2$ frames from the current window with $N_{\text{win}}/2$ frames from the corresponding prior window, predict the relative poses between loop edges from a single forward pass, and perform pose graph optimization on the $\mathrm{SE}(3)$ manifold to correct tracking drift.

During inference, LoGeR [Zhang et al., 2026] first tokenizes the current window of images with a DINO-based [Oquab et al., 2023] encoder, and then applies multiple attention layers to produce per-frame camera-to-world poses $\{\mathbf{T}_t\}_{t\in\mathcal{W}_k} \subset \mathrm{SE}(3)$ in a global world coordinate system and dense point maps $\mathbf{P}_t \in \mathbb{R}^{H\times W\times 3}$ in local camera coordinates.

To maintain global consistency over long sequences, LoGeR employs two mechanisms for cross-chunk information propagation. Sliding Window Attention (SWA) preserves uncompressed features from recent chunks and enables precise alignment at chunk boundaries, whereas Test-Time Training (TTT) fast weights accumulate an evolving global state that anchors the trajectory to a consistent coordinate frame and mitigates scale drift over long horizons. As a result, all per-frame poses $\{\mathbf{T}_t\}_{t=1}^{N}$ are expressed in a single global reference frame with consistent scale throughout inference.

Despite these properties, LoGeR still suffers from tracking drift over long trajectories, due to error accumulation caused by the lossy compression of global information in the TTT memory and the absence of explicit revisit constraints in the feedforward inference. This motivates us to build a loop closure framework on top of LoGeR. Two properties of the backbone make it particularly well suited as a frontend for loop closure: its chunk-based forward pass places no requirement on the temporal contiguity of input frames, and its pose outputs across chunks already share a common global scale. Exploiting these properties, we introduce a loop closure module on top of the LoGeR backbone, which estimates relative poses between looped frames under a consistent scale and enforces the resulting constraints through pose graph optimization.

3.2 Loop Detection and Alignment

To correct the tracking drift caused by error accumulation, we perform loop detection and loop pair alignment in a sequential manner.

Loop Detection. Let $\mathbf{I}_j$ denote a frame in the current window $\mathcal{W}^{\text{cur}}$ and $\mathbf{I}_i$ a frame from previous windows. Upon the arrival of a new window of N_{win} frames, we extract the compact global descriptor $\mathbf{d}_j^{\text{cur}}$ for each $\mathbf{I}_j$ via SALAD [Izquierdo and Civera, 2024], and compute its cosine similarity with

the descriptors $\mathbf{d}_i^{\text{prev}}$ of all frames from previous windows. A pair $(\mathbf{I}_j, \mathbf{I}_i)$ is considered a valid loop candidate if its similarity exceeds τ_{sim} and its temporal separation satisfies $j - i > \tau_{\text{time}}$. For each frame $\mathbf{I}_j$, we retain its top-k candidates ranked by similarity, yielding the loop pair set

$$\mathcal{C}_k = \bigcup_j \left\{ (\mathbf{I}_j, \mathbf{I}_i) \;\middle|\; i \in \text{Top}_k\big(s(\mathbf{d}_j^{\text{cur}}, \mathbf{d}_i^{\text{prev}})\big),\ s(\mathbf{d}_j^{\text{cur}}, \mathbf{d}_i^{\text{prev}}) > \tau_{\text{sim}},\ j - i > \tau_{\text{time}} \right\}, \tag{2}$$

where $s(\mathbf{d}_j^{\text{cur}}, \mathbf{d}_i^{\text{prev}}) = \dfrac{\mathbf{d}_j^{\text{cur}} \cdot \mathbf{d}_i^{\text{prev}}}{\|\mathbf{d}_j^{\text{cur}}\|_2 \, \|\mathbf{d}_i^{\text{prev}}\|_2}$ denotes the cosine similarity, and $\text{Top}_k(\cdot)$ selects the indices of the k frames with the largest similarity scores.

Loop-Conditioned Window Construction. The detected loop frames in $\mathcal{C}_k$ may come from several distinct prior windows. Accordingly, the current window can be associated with multiple prior windows, each yielding a separate loop-conditioned window. Formally, let $\{\mathcal{W}_m^{\text{prev}}\}_{m=1}^M$ denote the set of prior windows connected to the current window $\mathcal{W}^{\text{cur}}$ through loop pairs. For each $\mathcal{W}_m^{\text{prev}}$, we denote the set of loop pairs linking it to $\mathcal{W}^{\text{cur}}$ as:

$$\mathcal{L}_m = \{(\mathbf{I}_j, \mathbf{I}_i) \in \mathcal{C}_k \mid \mathbf{I}_j \in \mathcal{W}^{\text{cur}},\ \mathbf{I}_i \in \mathcal{W}_m^{\text{prev}}\}. \tag{3}$$

We then construct two subsets $\mathcal{S}_m^{\text{cur}} \subseteq \mathcal{W}^{\text{cur}}$ and $\mathcal{S}_m^{\text{prev}} \subseteq \mathcal{W}_m^{\text{prev}}$, each of size $N_{\text{win}}/2$, from the frames that participate in $\mathcal{L}_m$. If more than $N_{\text{win}}/2$ frames are available on either side, we keep the $N_{\text{win}}/2$ frames with the highest similarity scores; if fewer are available, we augment the subset with temporally adjacent frames from the respective window. The loop-conditioned window is then formed as $\tilde{\mathcal{W}}_m = \mathcal{S}_m^{\text{cur}} \cup \mathcal{S}_m^{\text{prev}}$.

This yields a collection $\{\tilde{\mathcal{W}}_m\}_{m=1}^M$ of loop-conditioned windows that preserve local temporal continuity while explicitly incorporating long-range loop constraints.

Relative Pose Estimation. Each loop-conditioned window $\tilde{\mathcal{W}}_m$ is passed through the streaming inference model to obtain the poses. Unlike the standard sliding-window inference in Sec. 3.1, which operates on temporally contiguous frames, a single forward pass here jointly infers poses for frames from $\mathcal{W}^{\text{cur}}$ and $\mathcal{W}_m^{\text{prev}}$. Let $\mathbf{T}_j^{(m)}$ and $\mathbf{T}_i^{(m)}$ denote the poses of $\mathbf{I}_j \in \mathcal{W}^{\text{cur}}$ and $\mathbf{I}_i \in \mathcal{W}_m^{\text{prev}}$ estimated from $\tilde{\mathcal{W}}_m$. The relative pose constraint for each loop pair $(\mathbf{I}_j, \mathbf{I}_i) \in \mathcal{L}_m$ is

$$\mathbf{T}_{ji}^{\text{loop}} = \big(\mathbf{T}_j^{(m)}\big)^{-1} \mathbf{T}_i^{(m)}. \tag{4}$$

Collecting these constraints over all loop pairs and all $\{\tilde{\mathcal{W}}_m\}_{m=1}^M$ yields the inter-frame loop constraints used in the pose graph optimization (Sec. 3.3).

3.3 Pose Graph Optimization

Given sequential constraints between each frame and its N_{seq} temporal neighbors, and loop closure constraints $\{\mathbf{T}_{ji}^{\text{loop}}\}$ provided by Sec. 3.2, we jointly refine all frame poses via pose graph optimization on $\text{SE}(3)$, as LoGeR's SWA and TTT layers ensure a globally consistent scale. We denote the corresponding edge sets by $\mathcal{E}_{\text{seq}}$ and $\mathcal{E}_{\text{loop}}$, and use $\mathbf{T}_{qp}^{\text{seq}}$ to represent the relative pose on a sequential edge $(q, p) \in \mathcal{E}_{\text{seq}}$. Thus, we formulate the following cost function to optimize the global poses $\{\mathbf{T}_t\}_{t=1}^N \subset \text{SE}(3)$:

$$\begin{aligned}\{\mathbf{T}_t^*\} = \arg\min_{\{\mathbf{T}_t\}} & \sum_{(q,p)\in\mathcal{E}_{\text{seq}}} \left\| \text{Log}_{\text{SE}(3)}\big((\mathbf{T}_{qp}^{\text{seq}})^{-1} \mathbf{T}_q^{-1} \mathbf{T}_p\big) \right\|_2^2 \\ & + \sum_{(j,i)\in\mathcal{E}_{\text{loop}}} \rho_\delta\left(\left\| \text{Log}_{\text{SE}(3)}\left((\mathbf{T}_{ji}^{\text{loop}})^{-1} \mathbf{T}_j^{-1} \mathbf{T}_i\right) \right\| \right),\end{aligned} \tag{5}$$

where $\text{Log}_{\text{SE}(3)}(\cdot) : \text{SE}(3) \to \mathbb{R}^6$ maps a transformation to its 6-dimensional tangent vector, and $\rho_\delta(\cdot)$ is the Huber loss with threshold δ, applied to the loop term to suppress the influence of outlier loop constraints as follows:

$$\rho_\delta(r) = \begin{cases} \frac{1}{2} r^2, & |r| \le \delta, \\ \delta\left(|r| - \frac{1}{2}\delta\right), & |r| > \delta. \end{cases} \tag{6}$$

We solve the problem with the Levenberg-Marquardt algorithm, and update the poses on the manifold via $\mathbf{T}_t \leftarrow \mathbf{T}_t \exp(\boldsymbol{\xi}_t^\wedge)$ at each iteration, where $\boldsymbol{\xi}_t \in \mathbb{R}^6$ is the tangent-space increment and $(\cdot)^\wedge$ denotes the hat operator that lifts $\mathbb{R}^6$ to $\mathfrak{se}(3)$.

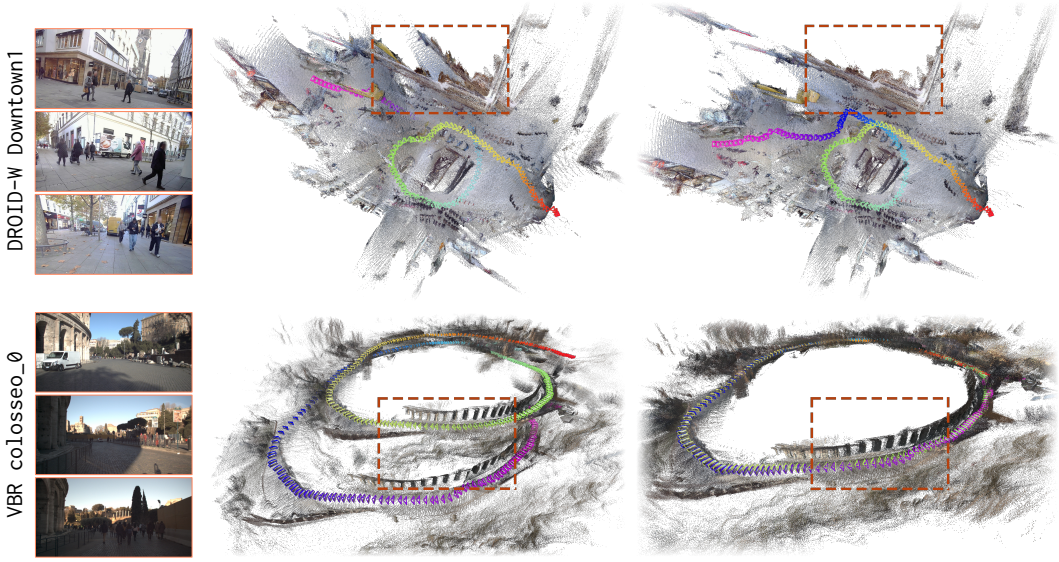

Figure 3: **Reconstruction comparison.** LoGeR [Zhang et al., 2026] fails to close loops and produces misaligned duplicate geometry, whereas our method recovers globally consistent reconstructions. Key regions are highlighted with red dashed boxes.

4 Experiments

Benchmarks and Metrics. We evaluate our method on (a) three representative long-context reconstruction benchmarks: the VBR dataset [Brizi et al., 2024], the autonomous driving benchmark KITTI Odometry [Geiger et al., 2013], and the challenging Oxford Spires dataset [Tao et al., 2024]; and (b) one outdoor dynamic benchmark, DROID-W [Li et al., 2026]. VBR is captured in Rome and provides RGB sequences, 3D point clouds, IMU measurements, and ground-truth trajectories obtained via bundle adjustment. Following LoGeR, we evaluate on 7 VBR sequences ranging from 8,815 to 18,846 frames and 1.45 to 5.2 km in trajectory length, all containing loops. KITTI Odometry comprises 11 sequences with long, loop-containing trajectories of up to 4,661 frames and 5.1 km. Oxford Spires consists of 14 sequences spanning indoor and outdoor scenes with challenging loop closures, ranging from 3,961 to 15,803 frames and 294 to 925 m in length. DROID-W contains 7 challenging dynamic sequences captured outdoors, ranging from 1,427 to 2,200 frames and 62 to 130 m in length. We report Absolute Trajectory Error (ATE) to evaluate tracking accuracy, and align estimated trajectories with the groundtruth via Sim(3) Umeyama alignment [Umeyama, 1991].

Baselines. We conduct extensive comparisons with both ***optimization-based*** and ***feedforward*** methods. For optimization-based methods, we compare against DROID-SLAM [Teed and Deng, 2021], DPV-SLAM / DPV-SLAM++ [Lipson et al., 2024], additionally including the dynamic SLAM methods WildGS-SLAM [Zheng et al., 2025] and DROID-W [Li et al., 2026] on the DROID-W dataset. For feedforward approaches, we compare against (a) *streaming reconstruction foundation models*: CUT3R [Wang et al., 2025b], TTT3R [Chen et al., 2025b], FastVGGT [Shen et al., 2025], InfiniteVGGT [Yuan et al., 2026], AMB3R-VO [Wang and Agapito, 2025], LingBot-Map [Chen et al., 2026], and LoGeR / LoGeR* [Zhang et al., 2026], where LoGeR* incorporates alignment-aware training and adopts a larger window size; (b) *VGGT-based SLAM*: VGGT-SLAM [Maggio et al., 2025, Maggio and Carlone, 2026] and VGGT-Long [Deng et al., 2025], for which we adopt their default chunk sizes of 16 and 60, respectively. Following prior work [Zhang et al., 2026], we additionally evaluate π^3 [Wang et al., 2025c] in a chunk-wise manner.

Quantitative Results. Camera tracking comparisons on four benchmarks are reported in Tables 1, 2, 3, and 4. As shown in Table 1 and Table 2, both VBR and KITTI consist of kilometer-scale sequences, on which our approach substantially outperforms LoGeR / LoGeR* [Zhang et al., 2026], as our loop closure module effectively handles revisited regions. LoGeR* achieves the second-best performance due to its alignment-aware training and large window size. Prior VGGT-based alignment methods exhibit larger tracking errors due to inaccurate point cloud alignment, and recent feedforward streaming models suffer from noticeable drift. We further compare against classical deep visual

Table 1: **Comparison of absolute trajectory error** (ATE RMSE ↓ [m]) on VBR [Brizi et al., 2024]. The top and bottom blocks denote *optimization-based* and *feedforward* methods, respectively. Optimization-based methods are shown in gray, with the best result per sequence in **bold**. Among feedforward methods, best results are highlighted as **first** , second , and third .

Method	colosseo_0	campus_0	campus_1	pincio_0	spagna_0	diag_0	ciampino_1	**Avg.**
#Frames	8815	12042	11671	11142	14141	10021	18846	12383
Scale	1.45km	2.73km	2.95km	1.27km	1.56km	1.02km	5.20km	2.31km
Contains Loop?	✓	✓	✓	✓	✓	✓	✓	–
Optimization-based (Calibrated)								
DROID-SLAM	76.53	128.77	63.89	**43.86**	50.63	**35.58**	132.84	76.01
DPV-SLAM	107.55	56.85	88.46	102.64	**32.63**	74.11	60.95	74.74
DPV-SLAM++	**55.47**	**11.90**	**13.11**	101.84	34.12	74.69	**59.47**	**50.09**
Feedforward (Uncalibrated)								
VGGT-Long	45.73	132.68	115.65	64.40	58.49	33.66	187.97	91.23
VGGT-SLAM	102.92	110.77	89.69	72.98	62.67	35.62	144.02	88.38
VGGT-SLAM 2.0	102.25	26.57	22.42	47.05	59.91	36.00	51.53	49.39
InfiniteVGGT	90.02	136.83	116.69	77.72	63.47	33.91	202.11	102.96
CUT3R	88.16	46.39	48.00	52.87	47.98	30.18	187.58	71.59
TTT3R	81.63	68.36	65.67	35.56	39.03	19.84	185.59	70.81
AMB3R	62.60	110.60	119.76	61.69	62.04	35.02	185.44	91.02
π^3-Chunk	81.63	86.54	71.49	49.85	54.27	28.07	118.81	70.09
LingBot-Map	19.22	18.46	11.40	41.89	36.56	22.65	67.64	31.12
LoGeR	25.60	19.53	28.90	28.11	23.04	35.01	92.94	36.16
LoGeR*	49.38	22.44	34.90	11.08	27.05	32.88	44.51	31.75
CLoSeR (Ours)	**14.23**	**7.43**	**3.71**	**9.78**	**13.04**	**12.92**	**16.93**	**11.15**

Table 2: **Comparison of absolute trajectory error** (ATE RMSE ↓ [m]) on KITTI [Geiger et al., 2013]. Traditional deep visual SLAM systems suffer from scale drift under KITTI's forward-dominant motion. Our approach outperforms all baselines.

Method	00	01	02	03	04	05	06	07	08	09	10	**Avg.**
#Frames	4541	1101	4661	801	271	2761	1101	1101	4071	1591	1201	2109
Scale	3.7km	2.5km	5.1km	0.6km	0.4km	2.2km	1.2km	0.7km	3.2km	1.7km	0.9km	2.0km
Contains Loop?	✓	×	✓	×	×	✓	✓	✓	×	✓	×	–
Optimization-based (Calibrated)												
DROID-SLAM	92.10	344.60	107.61	**2.38**	1.00	118.50	62.47	21.78	161.60	**72.32**	118.70	100.28
DPV-SLAM	112.80	**11.50**	123.53	2.50	0.81	57.80	54.86	18.77	**110.49**	76.66	**13.65**	53.03
DPV-SLAM++	**8.30**	11.86	**39.64**	2.50	**0.78**	**5.74**	**11.60**	**1.52**	110.90	76.70	13.70	**25.75**
Feedforward (Uncalibrated)												
VGGT-Long	8.67	121.17	32.08	6.12	4.23	8.31	5.34	**4.63**	53.10	41.99	18.37	27.64
VGGT-SLAM 2.0	108.39	155.00	140.15	13.45	2.87	10.12	29.17	17.27	75.33	27.70	24.09	54.87
FastVGGT	OOM	639.39	OOM	21.53	9.51	OOM	40.56	51.35	OOM	201.54	196.22	–
InfiniteVGGT	186.46	623.62	289.16	166.74	68.00	143.84	117.57	85.33	221.56	215.41	156.92	206.78
CUT3R	190.38	90.59	264.39	20.40	7.31	92.25	67.54	22.48	145.08	67.42	40.00	91.62
TTT3R	119.94	99.59	238.07	16.83	3.98	36.38	47.20	11.62	107.33	86.96	33.58	72.86
AMB3R	177.78	244.67	233.65	22.08	6.81	157.62	61.93	35.02	255.53	134.62	50.41	125.47
π^3-Chunk	26.65	196.04	157.92	5.13	1.09	12.79	27.66	5.94	61.26	56.31	21.96	52.07
LingBot-Map	23.22	88.84	71.15	**2.81**	**0.91**	17.40	4.91	5.16	**14.24**	21.71	14.54	24.08
LoGeR	62.34	**41.64**	39.64	4.89	1.82	41.27	13.99	16.24	26.46	22.71	8.84	25.44
LoGeR*	30.47	47.91	36.32	5.38	1.95	26.34	6.60	5.55	24.41	10.12	10.11	18.65
CLoSeR (Ours)	**6.58**	41.80	**28.38**	4.88	1.82	**6.58**	**4.77**	4.87	26.51	**5.64**	**8.83**	**12.79**

SLAM systems. DROID-SLAM [Teed and Deng, 2021] and DPV-SLAM [Lipson et al., 2024] often fall behind in outdoor kilometer-scale scenes under monocular input, where scale drift and forward-dominant motion (e.g., KITTI Odometry) jointly degrade tracking. In contrast, our approach remains robust, benefiting from strong 3D reconstruction priors and effective loop closure. DROID-W dataset contains strong dynamic motion (e.g., moving crowds), where dynamics-aware methods such as DROID-W [Li et al., 2026] unsurprisingly perform best. However, our method substantially outperforms all other baselines and, notably, remains competitive with in-the-wild methods despite not incorporating any dynamics-specific handling. Table 4 further highlights the superiority of our approach on the challenging Oxford Spires sequences.

Qualitative Comparisons. We provide qualitative point cloud comparisons in Fig. 3. On `colosseo_0`, LoGeR reconstructs the Colosseum as two misaligned duplicates (red boxes), revealing its failure to close the loop upon revisit. In contrast, our method recovers a single, globally consistent structure. The same failure pattern appears on `Downtown1`, where LoGeR yields fragmented geometry around the loop region whereas ours produces a coherent reconstruction. Fig. 4 compares estimated trajectories on four benchmarks. VGGT-SLAM and VGGT-Long both construct submaps with VGGT and

Table 3: **Comparison of absolute trajectory error** (ATE RMSE ↓ [m]) on DROID-W [Li et al., 2026]. Optimization-based methods are shown in gray, with the best result per sequence in **bold**. Among feedforward methods, best results are highlighted as **first**, second, and third. Our approach yields robust and accurate camera pose estimation on dynamic sequences, despite not explicitly handling dynamic objects.

Method	downtown1	downtown2	downtown3	downtown4	downtown5	downtown6	downtown7	**Avg.**
#Frames	1427	2200	1438	1794	2157	1900	1900	1831
Scale	91m	122m	62m	85m	130m	105m	109m	101m
Contains Loop?	✓	✓	✓	✓	×	✓	✓	–
Optimization-based (Calibrated)								
DROID-SLAM	0.26	7.84	1.05	0.33	0.64	**0.06**	**0.05**	1.46
DPV-SLAM	0.41	0.50	1.05	**0.15**	1.05	7.55	7.95	2.67
DPV-SLAM++	0.41	**0.12**	**0.15**	0.16	1.06	2.57	7.04	1.64
WildGS-SLAM	**0.10**	0.95	0.43	0.36	0.87	1.22	0.53	0.63
DROID-W	0.15	0.25	**0.15**	0.32	**0.24**	0.43	0.07	**0.23**
Feedforward (Uncalibrated)								
VGGT-Long	4.68	4.94	2.48	5.11	4.61	3.14	1.79	3.82
VGGT-SLAM 2.0	0.47	2.58	2.03	4.36	4.82	**0.42**	1.30	2.28
CUT3R	10.04	11.91	6.54	9.70	21.78	7.74	5.85	10.51
TTT3R	4.64	11.25	4.30	7.35	11.28	5.09	7.26	7.31
AMB3R	1.90	5.72	5.57	4.99	7.65	7.69	6.59	5.73
LingBot-Map	1.00	**0.70**	**0.46**	**0.74**	1.37	1.63	0.45	0.91
LoGeR	2.16	1.98	1.02	1.78	0.74	1.64	0.73	1.44
CLoSeR (Ours)	**0.34**	0.88	0.99	1.52	**0.73**	0.50	**0.30**	**0.75**

Table 4: **Comparison of absolute trajectory error** (ATE RMSE ↓ [m]) on Oxford Spires [Tao et al., 2024]. Our approach achieves the best pose estimation performance on this challenging dataset, demonstrating the superiority and robustness of our method.

Method	ke-02	ke-03	ke-04	ke-05	ob-01	ob-02	bl-01	bl-02	bl-05	ch-01	ch-02	ch-03	ch-05	bo-02	**Avg.**
#Frames	5969	5682	13518	11544	5746	5469	8041	7276	6755	15803	11775	6204	3961	4966	8051
Scale	294m	283m	783m	706m	397m	393m	494m	393m	386m	925m	642m	339m	823m	691m	539m
Contains Loop?	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	×	–
Optimization-based (Calibrated)															
DROID-SLAM	6.66	29.71	**14.10**	36.60	**0.17**	**0.36**	47.79	6.05	15.88	13.85	10.17	0.33	**8.92**	5.64	14.02
DPV-SLAM	5.84	**4.53**	19.74	25.25	0.27	6.24	**3.67**	**0.89**	**11.62**	**1.05**	**1.15**	0.94	42.62	3.51	**9.10**
DPV-SLAM++	**1.33**	29.35	18.39	**3.66**	0.20	2.94	3.88	1.00	12.06	3.93	7.82	**0.28**	40.19	**3.32**	9.17
Feedforward (Uncalibrated)															
VGGT-Long	3.63	14.00	32.66	**4.64**	3.80	5.64	11.92	14.04	11.41	17.59	9.03	4.02	14.51	16.76	11.69
VGGT-SLAM 2.0	7.28	20.57	6.90	17.06	**1.76**	**1.85**	18.74	**1.23**	**1.26**	**1.50**	**1.64**	**0.96**	16.41	**8.51**	7.55
CUT3R	25.44	17.75	42.51	36.12	26.83	27.05	50.64	42.53	49.45	33.46	41.67	11.83	32.47	56.71	35.32
TTT3R	29.05	17.45	32.54	41.37	26.37	26.65	51.27	45.89	35.25	31.17	41.46	14.18	27.26	69.42	34.95
AMB3R	15.95	29.38	24.68	40.05	2.99	3.36	47.88	41.59	42.97	28.91	43.34	14.56	26.64	18.79	27.22
LingBot-Map	3.42	3.27	**3.00**	9.39	2.81	2.50	18.95	3.82	2.78	3.32	13.39	2.94	28.79	9.27	7.69
LoGeR	4.01	3.11	6.51	5.76	7.01	6.98	**6.58**	7.33	6.98	9.18	6.72	1.84	10.22	14.09	6.88
CLoSeR (Ours)	**2.75**	**2.40**	5.79	4.65	4.33	4.03	7.78	5.18	5.06	2.57	5.70	1.19	**8.03**	14.08	**5.26**

align submaps via point cloud registration on overlapping frames. However, independent per-submap predictions and noisy point clouds lead to inaccurate alignment and unreliable scale, causing substantial tracking errors. LingBot-Map [Chen et al., 2026] produces noisy trajectories with substantial jitter, while LoGeR produces erroneous poses, especially in revisited regions due to large tracking drift. In contrast, our method delivers the most accurate and robust trajectories estimation across these sequences, benefiting from its effective loop closure mechanism.

Ablation Study. We report ablations of several design choices in Table 5. In variant (b), we set $N_{\text{seq}} = 1$ to enforce neighbor-only sequential constraints; the resulting performance drop shows that denser sequential constraints provide stronger short-range regularization of the trajectory. In variant (c), removing the Huber loss degrades performance, confirming its effectiveness in down-weighting outlier loop constraints. In variant (d), we replace our SE(3) pose graph with a Sim(3) formulation that additionally optimizes per-frame scale. Since our loop closure window inherits scale consistency from the streaming backbone, the extra scale degrees of freedom introduce optimization ambiguity and lead to inferior accuracy. In variant (e), we further replace it with an SL(4) formulation proposed

Table 5: **Ablation study.** Average tracking error on the VBR [Brizi et al., 2024] dataset. We ablate five components of our method and compare against the full model.

Method	ATE RMSE [m] ↓
(a) LoGeR (w/o Loop Closure)	36.16
(b) Neighbor-only constr.	14.25
(c) w/o Huber loss	14.15
(d) Sim(3) optimization	13.33
(e) SL(4) optimization	20.36
Full method	**11.15**

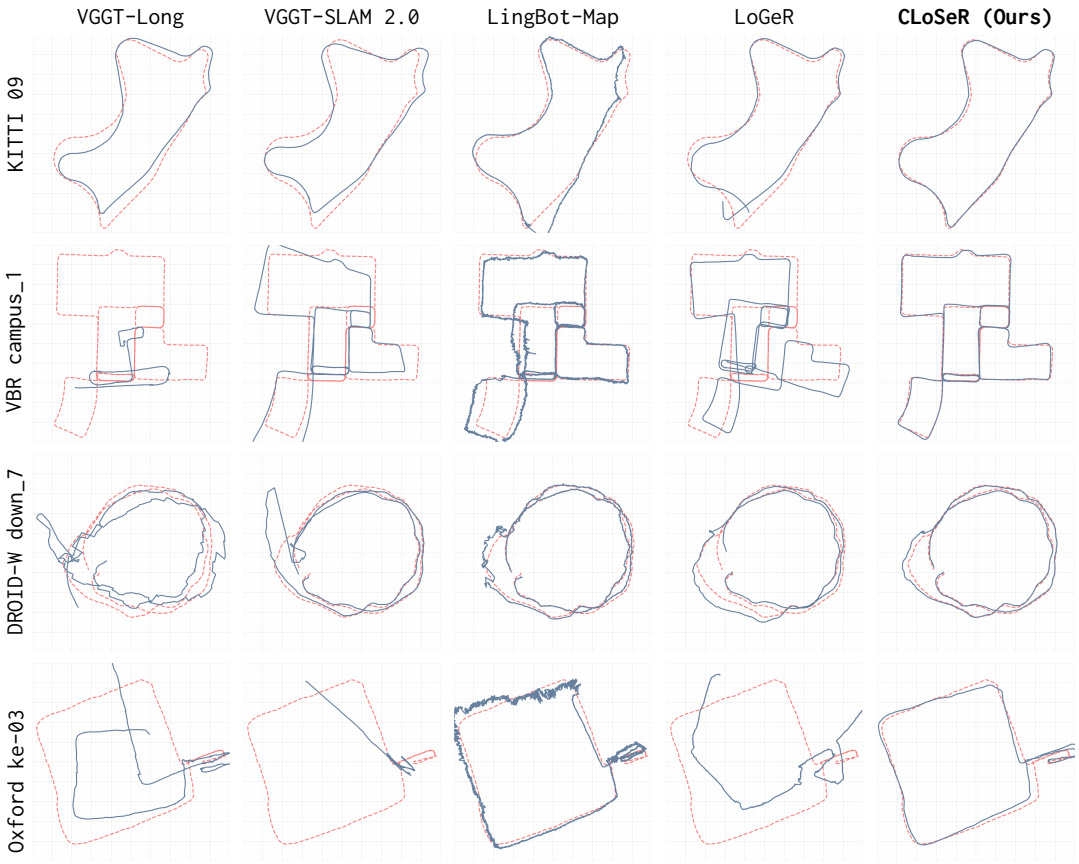

Figure 4: **Trajectory comparison** on KITTI Odometry [Geiger et al., 2013], VBR [Brizi et al., 2024], DROID-W [Li et al., 2026], and Oxford Spires [Tao et al., 2024] datasets. Groundtruth trajectories (coral, dashed) are overlaid with the estimates from each method (gray-blue, solid) after Sim(3) Umeyama alignment [Umeyama, 1991].

in VGGT-SLAM, whose 15-DOF projective transformation introduces more degrees of freedom than necessary in our setting, resulting in additional performance degradation. Overall, our full method achieves the best performance, validating the effectiveness of each design choice.

5 Limitations

We present a feedforward streaming reconstruction method that incorporates loop closure to substantially reduce drift over kilometer-scale trajectories. However, when the underlying odometry suffers not from accumulated drift but from complete failure, our method cannot recover the correct trajectory. Such failures arise when the feedforward frontend (LoGeR in our case) is pushed beyond its operating conditions, e.g., in highly dynamic environments or under aggressive camera motion.

6 Conclusion

We presented **CLoSeR**, a streaming reconstruction framework that revisits loop closure within a feedforward backbone, enabling accurate kilometer-scale reconstruction from uncalibrated monocular videos without any retraining. By exploiting the globally consistent scale of the streaming backbone LoGeR, we introduce a loop-conditioned window mechanism that recovers relative poses between looped frames directly in the global coordinate system. Our experiments demonstrate that pose graph optimization on the SE(3) manifold is sufficient within our framework, avoiding the Sim(3) or SL(4) optimization adopted by prior submap-based methods. In future work, we aim to improve the robustness of our framework under challenging conditions such as motion blur, low-texture regions, and exposure variations. We also plan to explore multi-agent collaborative reconstruction and online deployment, broadening its applicability to real-world large-scale environments.

References

Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef 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.

Philip Arm, Gabriel Waibel, Jan Preisig, Turcan Tuna, Ruyi Zhou, Valentin Bickel, Gabriela Ligeza, Takahiro Miki, Florian Kehl, Hendrik Kolvenbach, et al. Scientific exploration of challenging planetary analog environments with a team of legged robots. *Science robotics*, 8(80):eade9548, 2023.

Claudine Badue, Rânik Guidolini, Raphael Vivacqua Carneiro, Pedro Azevedo, Vinicius B Cardoso, Avelino Forechi, Luan Jesus, Rodrigo Berriel, Thiago M Paixao, Filipe Mutz, et al. Self-driving cars: A survey. *Expert systems with applications*, 165:113816, 2021.

Eric Brachmann, Jamie Wynn, Shuai Chen, Tommaso Cavallari, Aron Monszpart, Daniyar Turmukhambetov, and Victor Adrian Prisacariu. Scene coordinate reconstruction: Posing of image collections via incremental learning of a relocalizer. In *European Conference on Computer Vision*, pages 421–440. Springer, 2024.

Leonardo Brizi, Emanuele Giacomini, Luca Di Giammarino, Simone Ferrari, Omar Salem, Lorenzo De Rebotti, and Giorgio Grisetti. Vbr: A vision benchmark in rome. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 15868–15874. IEEE, 2024.

Carlos Campos, Richard Elvira, Juan J Gómez Rodríguez, José MM Montiel, and Juan D Tardós. Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam. *IEEE transactions on robotics*, 37(6):1874–1890, 2021.

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

Xingyu Chen, Yue Chen, Yuliang Xiu, Andreas Geiger, and Anpei Chen. Easi3r: Estimating disentangled motion from dust3r without training. *arXiv preprint arXiv:2503.24391*, 2025a.

Xingyu Chen, Yue Chen, Yuliang Xiu, Andreas Geiger, and Anpei Chen. Ttt3r: 3d reconstruction as test-time training. *arXiv preprint arXiv:2509.26645*, 2025b.

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.

Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 224–236, 2018.

Mihai Dusmanu, Ignacio Rocco, Tomas Pajdla, Marc Pollefeys, Josef Sivic, Akihiko Torii, and Torsten Sattler. D2-net: A trainable cnn for joint detection and description of local features. *arXiv preprint arXiv:1905.03561*, 2019.

Johan Edstedt, Qiyu Sun, Georg Bökman, Mårten Wadenbäck, and Michael Felsberg. Roma: Robust dense feature matching. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 19790–19800, 2024.

Sven Elflein, Ruilong Li, Sérgio Agostinho, Zan Gojcic, Laura Leal-Taixé, Qunjie Zhou, and Aljosa Osep. Vgg-t^3: Offline feed-forward 3d reconstruction at scale. *arXiv preprint arXiv:2602.23361*, 2026.

Jakob Engel, Thomas Schöps, and Daniel Cremers. Lsd-slam: Large-scale direct monocular slam. In *European conference on computer vision*, pages 834–849. Springer, 2014.

Jakob Engel, Vladlen Koltun, and Daniel Cremers. Direct sparse odometry. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 2017.

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

Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The international journal of robotics research*, 32(11):1231–1237, 2013.

Sergio Izquierdo and Javier Civera. Optimal transport aggregation for visual place recognition. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, pages 17658–17668, 2024.

Haian Jin, Rundi Wu, Tianyuan Zhang, Ruiqi Gao, Jonathan T Barron, Noah Snavely, and Aleksander Holynski. Zipmap: Linear-time stateful 3d reconstruction with test-time training. *arXiv preprint arXiv:2603.04385*, 2026.

Michael Kaess, Hordur Johannsson, Richard Roberts, Viorela Ila, John J Leonard, and Frank Dellaert. isam2: Incremental smoothing and mapping using the bayes tree. *The International Journal of Robotics Research*, 31 (2):216–235, 2012.

Nikhil Keetha, Jay Karhade, Krishna Murthy Jatavallabhula, Gengshan Yang, Sebastian Scherer, Deva Ramanan, and Jonathon Luiten. Splatam: Splat track & map 3d gaussians for dense rgb-d slam. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Rainer Kümmerle, Giorgio Grisetti, Hauke Strasdat, Kurt Konolige, and Wolfram Burgard. g2o: A general framework for graph optimization. In *2011 IEEE international conference on robotics and automation*, pages 3607–3613. IEEE, 2011.

Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3d with mast3r. In *European conference on computer vision*, pages 71–91. Springer, 2024.

Stefan Leutenegger. Okvis2: Realtime scalable visual-inertial slam with loop closure. *arXiv preprint arXiv:2202.09199*, 2022.

Moyang Li, Zihan Zhu, Marc Pollefeys, and Daniel Barath. Droid-slam in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 36498–36508, June 2026.

Philipp Lindenberger, Paul-Edouard Sarlin, and Marc Pollefeys. Lightglue: Local feature matching at light speed. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 17627–17638, 2023.

Lahav Lipson, Zachary Teed, and Jia Deng. Deep patch visual slam. In *European Conference on Computer Vision*, pages 424–440. Springer, 2024.

Xu Liu, Jiuzhou Lei, Ankit Prabhu, Yuezhan Tao, Igor Spasojevic, Pratik Chaudhari, Nikolay Atanasov, and Vijay Kumar. Slideslam: Sparse, lightweight, decentralized metric-semantic slam for multi-robot navigation. *arXiv preprint arXiv:2406.17249*, 2024.

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

Dominic Maggio, Hyungtae Lim, and Luca Carlone. Vggt-slam: Dense rgb slam optimized on the sl (4) manifold. *arXiv preprint arXiv:2505.12549*, 2025.

Hidenobu Matsuki, Riku Murai, Paul HJ Kelly, and Andrew J Davison. Gaussian splatting slam. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Raul Mur-Artal and Juan D Tardós. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE transactions on robotics*, 2017.

Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. Orb-slam: a versatile and accurate monocular slam system. *IEEE transactions on robotics*, 2015.

Richard A Newcombe, Steven J Lovegrove, and Andrew J Davison. Dtam: Dense tracking and mapping in real-time. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, pages 2320–2327. IEEE, 2011.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

Linfei Pan, Dániel Baráth, Marc Pollefeys, and Johannes L Schönberger. Global structure-from-motion revisited. In *European Conference on Computer Vision*, pages 58–77. Springer, 2024.

Rémi Pautrat, Iago Suárez, Yifan Yu, Marc Pollefeys, and Viktor Larsson. Gluestick: Robust image matching by sticking points and lines together. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9706–9716, 2023.

Tong Qin, Peiliang Li, and Shaojie Shen. Vins-mono: A robust and versatile monocular visual-inertial state estimator. *IEEE transactions on robotics*, 34(4):1004–1020, 2018.

Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4938–4947, 2020.

J. L. Schönberger and J. M. Frahm. Structure-from-motion revisited. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016.

Tixiao Shan, Brendan Englot, Drew Meyers, Wei Wang, Carlo Ratti, and Daniela Rus. Lio-sam: Tightly-coupled lidar inertial odometry via smoothing and mapping. In *2020 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 5135–5142. IEEE, 2020.

You Shen, Zhipeng Zhang, Yansong Qu, Xiawu Zheng, Jiayi Ji, Shengchuan Zhang, and Liujuan Cao. Fastvggt: Training-free acceleration of visual geometry transformer. *arXiv preprint arXiv:2509.02560*, 2025.

Noah Snavely, Steven M Seitz, and Richard Szeliski. Modeling the world from internet photo collections. *International journal of computer vision*, 80(2):189–210, 2008.

Yu Sun, Xinhao Li, Karan Dalal, Jiarui Xu, Arjun Vikram, Genghan Zhang, Yann Dubois, Xinlei Chen, Xiaolong Wang, Sanmi Koyejo, et al. Learning to (learn at test time): Rnns with expressive hidden states. *arXiv preprint arXiv:2407.04620*, 2024.

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*, page 02783649251369905, 2024.

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

Zachary Teed and Jia Deng. Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Zachary Teed, Lahav Lipson, and Jia Deng. Deep patch visual odometry. *Advances in Neural Information Processing Systems*, 36:39033–39051, 2023.

Shinji Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 1991.

Hengyi Wang and Lourdes Agapito. Amb3r: Accurate feed-forward metric-scale 3d reconstruction with backend. *arXiv preprint arXiv:2511.20343*, 2025.

Jianyuan Wang, Nikita Karaev, Christian Rupprecht, and David Novotny. Vggsfm: Visual geometry grounded deep structure from motion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 21686–21697, 2024a.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 5294–5306, 2025a.

Qianqian Wang, Yifei Zhang, Aleksander Holynski, Alexei A Efros, and Angjoo Kanazawa. Continuous 3d perception model with persistent state. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 10510–10522, 2025b.

Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2024b.

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.

Xingkui Wei, Yinda Zhang, Zhuwen Li, Yanwei Fu, and Xiangyang Xue. Deepsfm: Structure from motion via deep bundle adjustment. In *European conference on computer vision*, pages 230–247. Springer, 2020.

Wei Xu and Fu Zhang. Fast-lio: A fast, robust lidar-inertial odometry package by tightly-coupled iterated kalman filter. *IEEE Robotics and Automation Letters*, 6(2):3317–3324, 2021.

Chi Yan, Delin Qu, Dan Xu, Bin Zhao, Zhigang Wang, Dong Wang, and Xuelong Li. Gs-slam: Dense visual slam with 3d gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19595–19604, 2024.

Kwang Moo Yi, Eduard Trulls, Vincent Lepetit, and Pascal Fua. Lift: Learned invariant feature transform. In *European conference on computer vision*, pages 467–483. Springer, 2016.

Shuai Yuan, Yantai Yang, Xiaotian Yang, Xupeng Zhang, Zhonghao Zhao, Lingming Zhang, and Zhipeng Zhang. Infinitevggt: Visual geometry grounded transformer for endless streams. *arXiv preprint arXiv:2601.02281*, 2026.

Ji Zhang, Sanjiv Singh, et al. Loam: Lidar odometry and mapping in real-time. In *Robotics: Science and systems*, volume 2, pages 1–9. Berkeley, CA, 2014.

Junyi Zhang, Charles Herrmann, Junhwa Hur, Varun Jampani, Trevor Darrell, Forrester Cole, Deqing Sun, and Ming-Hsuan Yang. Monst3r: A simple approach for estimating geometry in the presence of motion. *arXiv preprint arxiv:2410.03825*, 2024.

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*, 2026.

Tianyuan Zhang, Sai Bi, Yicong Hong, Kai Zhang, Fujun Luan, Songlin Yang, Kalyan Sunkavalli, William T Freeman, and Hao Tan. Test-time training done right. *arXiv preprint arXiv:2505.23884*, 2025.

Jianhao Zheng, Zihan Zhu, Valentin Bieri, Marc Pollefeys, Songyou Peng, and Iro Armeni. Wildgs-slam: Monocular gaussian splatting slam in dynamic environments. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 11461–11471, 2025.

Xin Zhou, Xiangyong Wen, Zhepei Wang, Yuman Gao, Haojia Li, Qianhao Wang, Tiankai Yang, Haojian Lu, Yanjun Cao, Chao Xu, et al. Swarm of micro flying robots in the wild. *Science Robotics*, 7(66):eabm5954, 2022.

Zihan Zhu, Songyou Peng, Viktor Larsson, Weiwei Xu, Hujun Bao, Zhaopeng Cui, Martin R. Oswald, and Marc Pollefeys. Nice-slam: Neural implicit scalable encoding for slam. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Zihan Zhu, Songyou Peng, Viktor Larsson, Zhaopeng Cui, Martin R Oswald, Andreas Geiger, and Marc Pollefeys. Nicer-slam: Neural implicit scene encoding for rgb slam. In *Proc. of the International Conf. on 3D Vision (3DV)*, 2024.

Zihan Zhu, Wei Zhang, Moyang Li, Norbert Haala, Marc Pollefeys, and Daniel Barath. Vigs-slam: visual inertial gaussian splatting slam. In *European Conference on Computer Vision*, pages 322–341. Springer, 2026.

Dong Zhuo, Wenzhao Zheng, Jiahe Guo, Yuqi Wu, Jie Zhou, and Jiwen Lu. Streaming 4d visual geometry transformer. *arXiv preprint arXiv:2507.11539*, 2025.

A Technical appendices and supplementary material

A.1 Experimental Setting

We adopt *LoGeR* as our base model for all evaluations, as *LoGeR** uses different weights and a larger window size that incur substantially higher GPU memory consumption. In addition, *LoGeR** performs an SE(3) alignment across all window predictions to enforce a more consistent global reconstruction. We follow the evaluation protocol of LoGeR [Zhang et al., 2026] and downsample all input images at inference to a uniform pixel budget of approximately 2.55×10^5 pixels. This ensures fair comparison across datasets with different native resolutions. The resulting resolutions are 924×266 for KITTI Odometry [Geiger et al., 2013], 672×378 for VBR [Brizi et al., 2024], 672×378 for DROID-W [Li et al., 2026], and 588×434 for Oxford Spires [Tao et al., 2024].

We employ a window size of 32 and an overlap size of 3 across all evaluations, except for VBR, where we adopt a window size of 48 following the LoGeR protocol [Zhang et al., 2026]. For loop detection, we set the temporal separation threshold to $\tau_{\text{time}} = 4 \times$ window size and the similarity threshold to $\tau_{\text{sim}} = 0.7$, and retain only the top-5 most similar pairs. For pose graph optimization, we set $N_{\text{seq}} = 4$ to enforce stronger sequential constraints.

A.2 Additional Results

We provide additional qualitative comparisons of point clouds and camera trajectories in Fig. 5 and Fig. 6, respectively. As shown in Fig. 5, our approach produces more geometrically consistent reconstructions, since effective loop closure enforces constraints on revisited regions and enables drift-free reconstruction. Fig. 6 compares the estimated trajectories against baseline methods. VGGT-Long and VGGT-SLAM exhibit substantial tracking errors on the challenging VBR and Oxford Spires datasets, while LoGeR suffers from noticeable drift across all sequences due to the absence of explicit constraints on revisited regions. In contrast, our method robustly recovers accurate trajectories across all sequences.

Table 6: **Runtime and GPU memory analysis.** We report the average runtime breakdown (as a percentage of total runtime), the average number of windows, FPS, peak GPU memory usage, and trajectory accuracy on the KITTI [Geiger et al., 2013] dataset. PGO denotes pose graph optimization.

Method	Loop Detection	PGO	Model Inference	# Windows	FPS ↑	Peak GPU Mem. ↓	ATE RMSE [m] ↓
LoGeR [Zhang et al., 2026]	–	–	100.00%	73.1	**7.60**	16.65 GiB	25.44
CLoSeR (Ours)	5.16%	1.12%	93.72%	79.5	7.12	**15.69** GiB	**12.79**

A.3 Runtime and GPU Memory Analysis

We evaluate the runtime and GPU memory usage of our method compared with LoGeR Zhang et al. [2026]. All experiments are conducted on a single NVIDIA RTX 4090 GPU paired with 4 cores of an AMD EPYC 9554 CPU on a Slurm cluster. As shown in Table 6, our method incurs only marginal runtime overhead, achieving 7.12 FPS compared with 7.60 FPS for LoGeR (a 6.3% reduction in throughput), while reducing peak GPU memory usage from 16.65 GiB to 15.69 GiB (a 5.8% reduction) due to our efficient optimization. The additional cost mainly stems from our loop closure detection and the additional loop closure windows for model inference. In exchange, our method nearly halves trajectory error, reducing ATE RMSE from 25.44 m to 12.66 m – a substantial accuracy gain at marginal runtime cost.

Table 7: **Analysis of cross-window scale consistency.** We report average relative scale error on the KITTI [Geiger et al., 2013] dataset.

Method	Relative Scale Error Regular Window ↓	Relative Scale Error LC Window ↓
VGGT-long [Deng et al., 2025]	18.8%	49.3%
VGGT-SLAM 2.0 [Maggio and Carlone, 2026]	32.4%	147.2%
CLoSeR (Ours)	**5.8%**	**6.5%**

A.4 Analysis of Cross-Window Scale Consistency

We further analyze cross-window scale consistency for VGGT-Long [Deng et al., 2025], VGGT-SLAM 2.0 [Maggio and Carlone, 2026], and our CLoSeR. For each window (or submap) i, whether a regular window or a loop-closure one, we fit a $\mathrm{Sim}(3)$ transformation between its decoded pose estimates and the corresponding ground truth via Umeyama alignment Umeyama [1991], and extract the scalar scale s_i. For VGGT-SLAM 2.0, since its loop-closure submaps consist of only two frames, we instead use the segment-length ratio. We then define a sequence-level scale reference $s_{\mathrm{global}} = \frac{1}{|\mathcal{N}|}\sum_{i\in\mathcal{N}} s_i$, computed over the set of regular (non-loop-closure) windows $\mathcal{N}$, and compute the per-window relative scale error $e_i = |s_i/s_{\mathrm{global}} - 1|$. We report the scale error for both regular and loop-closure windows, averaged over all windows on KITTI Geiger et al. [2013]. As shown in Table 7, our method achieves substantially lower relative scale error for both regular and loop-closure windows. The improvement on regular windows benefits from LoGeR's long-context memory. The improvement on loop-closure windows comes from our loop-closure window construction: instead of relying on a separate relative pose estimator or feedforward module, it is naturally incorporated into the LoGeR streaming framework.

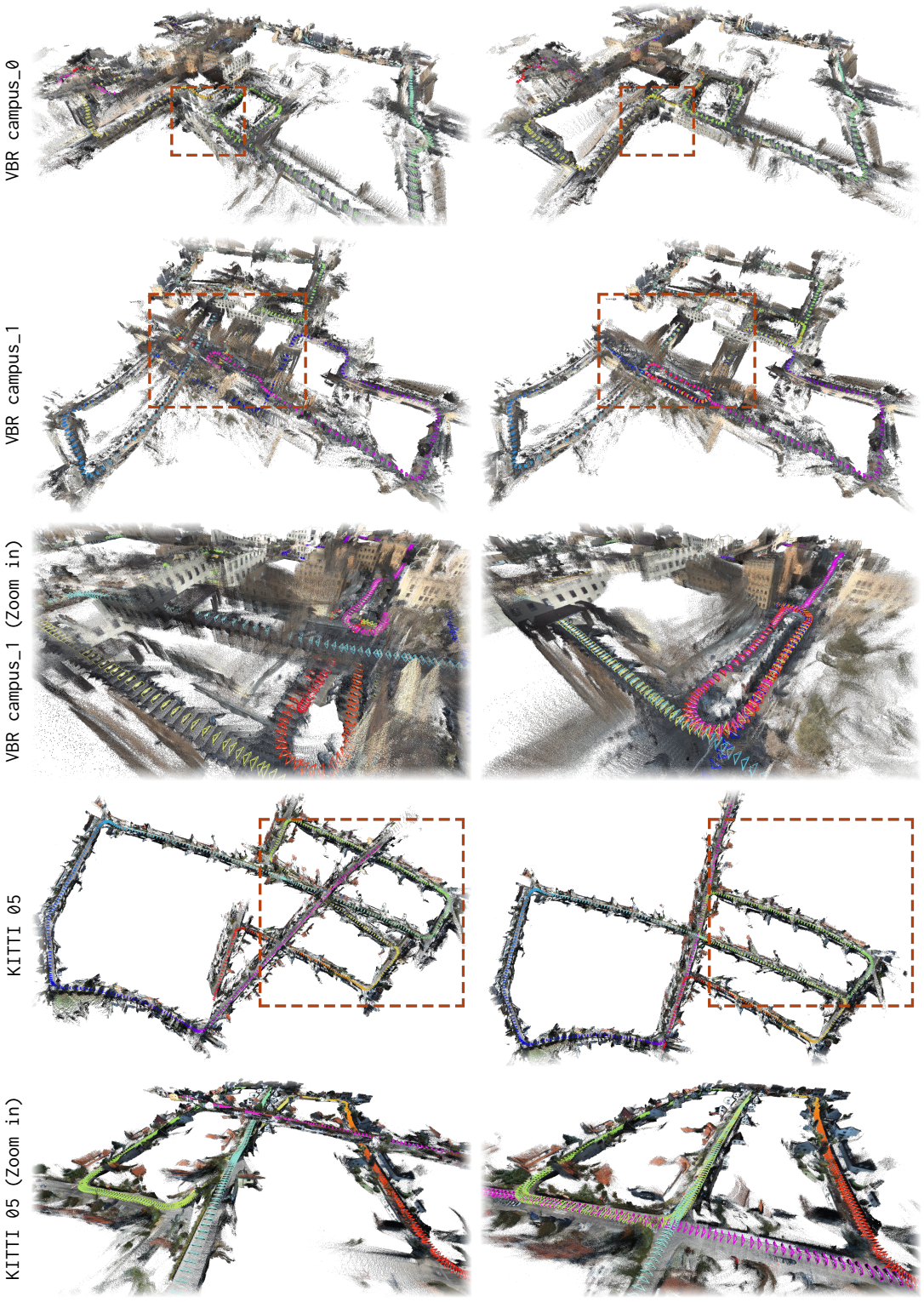

Figure 5: **Point clouds comparison.** Our method delivers more accurate geometry, while LoGeR exhibits noticeable ghosting artifacts caused by inaccurate pose estimation.

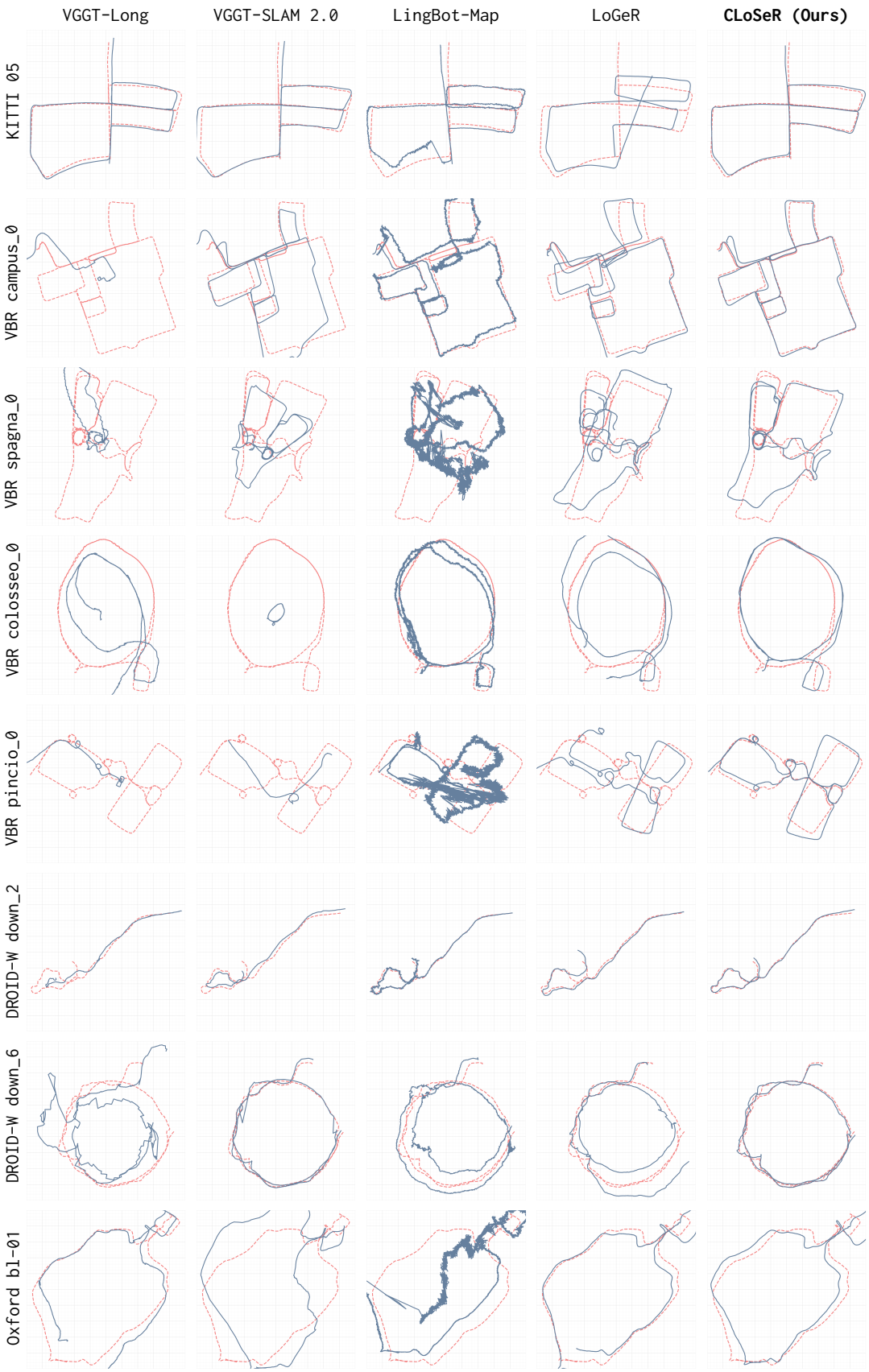

Figure 6: **Additional trajectory comparison.** Groundtruth trajectories (coral, dashed) are overlaid with the estimates from each method (gray-blue, solid) after Sim(3) Umeyama alignment [Umeyama, 1991]. Our approach robustly yields the most accurate trajectories across all sequences.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract (line 2-15) and introduction (Sec. 1) state three main contributions: (i) a system integrating loop closure into a feedforward streaming reconstruction model for kilometer-scale reconstruction; (ii) a loop-conditioned window mechanism that recovers relative poses between looped frames in a globally consistent coordinate frame; and (iii) the observation that SE(3) pose graph optimization suffices given the backbone's globally consistent scale, obviating the Sim(3) or SL(4) optimization used in prior submap-based methods. Each claim is substantiated by the method design in Sec. 3 and the experimental results in Sec. 4.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Limitations are explicitly discussed in limitations section (Sec. 5): our method corrects accumulated drift but cannot handle complete odometry failures of the LoGeR frontend, e.g. in highly dynamic environments or under aggressive camera motion. Future directions (motion blur, low texture, exposure variation, multi-agent) are also presented in conclusions (Sec. 6).

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.

- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

   Justification: This paper presents an empirical system rather than theoretical results. The mathematical formulations (Eqs. 1–6) describe the loop detection criterion, loop-conditioned constraints, and the SE(3) pose graph optimization objective. The paper does not introduce new theorems requiring proofs.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The method is described in detail in Sec. 3, covering loop detection, loop-conditioned window construction, and the SE(3) pose graph optimization formulation (Eqs. 1–6). Additional implementation details (window size, overlap size, N_{seq}, τ_{time}, and the per-dataset image resolutions) are reported in Appendix A.1. All evaluation benchmarks (VBR, KITTI Odometry, Oxford Spires, DROID-W) are publicly available, and we build on the publicly released LoGeR backbone without retraining. Code will be released.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.

- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: Code is not released at submission time but will be made public upon acceptance. The paper provides sufficient details to reproduce our results: Sec. 3 fully specifies the method (loop detection, loop-conditioned window construction, and SE(3) pose graph optimization in Eqs. 1-6), and Appendix A.1 reports all implementation details and hyperparameters. All evaluation benchmarks (VBR, KITTI Odometry, Oxford Spires, DROID-W) are publicly available. And we build on the publicly released LoGeR backbone without retraining.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Sec. 3 specifies the optimization details, including the Levenberg-Marquardt algorithm on the SE(3) manifold and the Huber-robustified loop term (Eqs. 5-6). Sec. 4

describes the benchmarks, evaluation metric (ATE RMSE with Sim(3) Umeyama alignment), baselines, and evaluation protocol. Appendix A.1 reports all remaining experimental details, such as hyperparameters.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: Our method is **deterministic**: we use a frozen pretrained LoGeR backbone, the loop detection and loop-conditioned window construction are deterministic given the inputs, and the SE(3) pose graph optimization is solved with Levenberg-Marquardt to convergence. Repeated runs therefore yield identical results, so error bars over multiple runs would be uninformative. Following standard practice in the SLAM and 3D reconstruction literature, we report single-run ATE RMSE per sequence (and the average across sequences) on the standard benchmark splits. Robustness is instead assessed across four diverse benchmarks (VBR, KITTI, Oxford Spires, DROID-W).

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: Appendix A.3 reports the compute setup (single NVIDIA RTX 4090 + 4 cores of AMD EPYC 9554 on a Slurm cluster), runtime (FPS), and peak GPU memory of our method compared with the LoGeR baseline.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.

- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: The research conforms to the NeurIPS Code of Ethics. The work uses publicly available benchmarks (see Sec. 4 for details) for evaluation, and does not involve human subjects, personal data, or content with potential for misuse beyond standard 3D reconstruction applications.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Positive: our method enables accurate kilometer-scale 3D reconstruction from monocular video, potentially lowering the cost of mapping pipelines that currently rely on expensive sensors (LiDAR, RTK-GPS), with applications in robotics and autonomous driving as discussed in Sec. 1. Negative: as foundational research on geometric scene reconstruction, the work has no direct path to negative societal applications beyond the general dual-use considerations shared by mapping and SLAM technologies.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not release pretrained generative models or scraped datasets that pose high risks of misuse. The proposed system performs camera pose estimation and 3D reconstruction from monocular videos using publicly available benchmarks (Sec. 4) and the pretrained LoGeR backbone (Sec. 3).

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All used assets are properly cited: the LoGeR backbone [Zhang et al., 2026], the SALAD descriptor [Izquierdo and Civera, 2024], and the benchmark datasets VBR [Brizi et al., 2024], KITTI [Geiger et al., 2013], Oxford Spires [Tao et al., 2024], and DROID-W [Li et al., 2026] (see Sec. 3 and Sec. 4). Baseline implementations are credited to their respective papers (Sec. 4).

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The paper does not introduce new datasets or pretrained models. The proposed method is built on top of an existing pretrained backbone (LoGeR) without retraining.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing or research with human subjects. All experiments are performed on publicly available SLAM benchmarks (see Sec. 4).

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing or research with human subjects, so IRB approval is not applicable.

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs were not used as any component of the core methodology. Our method augments a feedforward 3D reconstruction backbone with a loop closure module based on global descriptor retrieval and pose graph optimization - none of which involve LLMs.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.