

LoGeR: Long-Context Geometric Reconstruction with Hybrid Memory

Junyi Zhang^{1,2} Charles Herrmann^{1,*} Junhwa Hur^{1,*} Chen Sun¹ Ming-Hsuan Yang¹
Forrester Cole¹ Trevor Darrell² Deqing Sun^{1,†}

¹Google DeepMind ²UC Berkeley

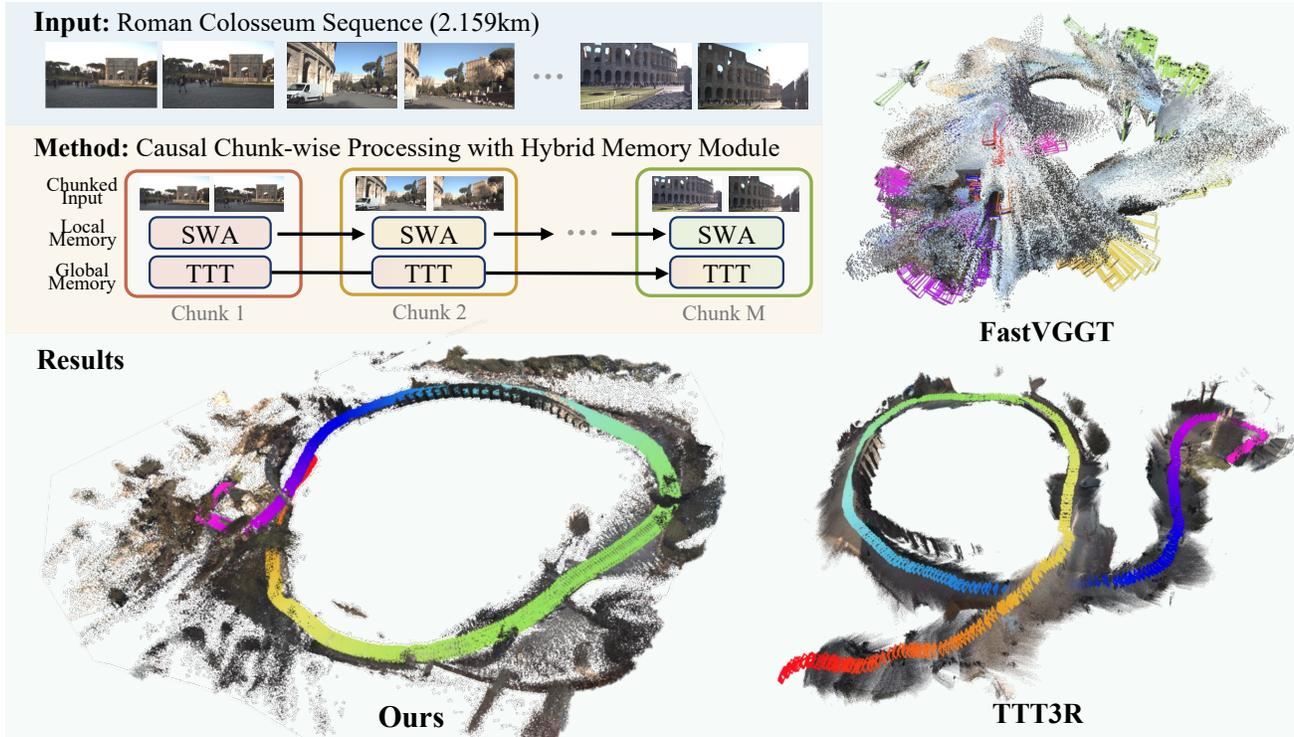


Figure 1. **Our proposed method and visual comparison.** For very long videos, we advocate chunk-based processing where bidirectional attention handles intra-chunk reasoning with inter-chunk alignment handled by our proposed *hybrid memory module*, composing Sliding Window Attention (SWA) for detailed local memory and Test-Time Training (TTT) for compressed global context. LoGeR shows clear improvement over prior methods on expansive VBR (Brizi et al., 2024) dataset, yielding superior loop closures and finer geometric details.

Abstract

Feedforward geometric foundation models achieve strong short-window reconstruction, yet scaling them to minutes-long videos is bottlenecked by quadratic attention complexity or limited effective memory in recurrent designs. We present **LoGeR (Long-context Geometric Reconstruction)**, a novel architecture that scales dense 3D reconstruction to extremely long sequences without post-optimization. LoGeR processes video streams in chunks, leveraging strong bidirectional priors for high-fidelity intra-chunk reasoning. To manage the critical challenge of coherence across chunk boundaries, we propose a learning-based *hybrid memory module*. This dual-component system com-

bins a parametric Test-Time Training (TTT) memory to anchor the global coordinate frame and prevent scale drift, alongside a non-parametric Sliding Window Attention (SWA) mechanism to preserve uncompressed context for high-precision adjacent alignment. Remarkably, this memory architecture enables LoGeR to be trained on sequences of 128 frames, and generalize up to thousands of frames during inference. Evaluated across standard benchmarks and a repurposed VBR dataset with sequences of up to 19k frames, LoGeR substantially outperforms prior state-of-the-art feedforward methods—reducing ATE on KITTI by over 74%—and achieves robust, globally consistent reconstruction over unprecedented horizons. Webpage: <https://LoGeR-project.github.io/>.

*Project leads, †Direction lead

1. Introduction

Large-scale dense 3D reconstruction has long been a central goal in computer vision, underpinning many important applications, from holistic scene understanding to generative world-building. For decades, classical optimization frameworks dominated this landscape; and while capable of reconstructing entire cities, they rely on computationally intensive offline processes and often falter on sparse or texture-less inputs. Recently, geometric foundation models—such as DUS_t3R (Wang et al., 2024), MonST3R (Zhang et al., 2025a), VGGT (Wang et al., 2025a), and π^3 (Wang et al., 2026)—are driving a paradigm shift. By distilling complex geometric priors from vast datasets, these models enable robust, feedforward inference even where classical methods fail. However, a critical gap remains. While classical pipelines scale to cities, feedforward models are currently confined to bounded scenes.

The primary obstacle to this scaling is twofold: a fundamental “context wall” inherent in current architectures and a severe “data wall” during training. Architecturally, while bidirectional attention is essential for learning complex geometric priors, its quadratic complexity restricts its use to short-context windows. From a data perspective, current models are predominantly trained on short-context “bubbles” (dozens to over a hundred frames), leaving them fundamentally ill-equipped to integrate long-range dependencies at inference time (thousands to tens of thousands of frames). Consequently, inference-time heuristics like FastVGGT (Shen et al., 2026), which successfully mitigate memory bottlenecks, still fail to generalize to the large-scale VBR dataset (Brizi et al., 2024), as shown in Figs. 1 and 3.

To overcome this data wall without waiting for the curation of massive long-horizon datasets with the requisite diversity and accuracy, we argue that end-to-end chunk-wise processing is a practical and effective strategy. Decomposing the sequence ensures that local inferences remain “in-distribution” relative to existing short-context training data and allows us to leverage strong bidirectional, attention-based baselines such as π^3 (Wang et al., 2026). However, this paradigm introduces a critical new challenge: managing memory and coherence across chunk boundaries. Achieving high-fidelity dense 3D reconstruction over long sequences requires balancing three distinct levels of coherence: intra-window details, uncompressed context for high-precision local alignment, and global structural integrity over long ranges. Existing sequential models fail to balance these conflicting needs. For instance, recurrent approaches like CUT3R (Wang et al., 2025b) compress all temporal context into a single lossy hidden state, sacrificing the high-precision dense information needed for seamless adjacent alignment. Conversely, naive deterministic stitching preserves local detail but lacks the long-range memory required to prevent scale drift.

Table 1. Architectural trade-offs in sequence modeling. Our hybrid memory module achieves a balance, preserving lossless local geometric details while maintaining global structural consistency at a linear computational cost with respect to sequence length.

Mechanism	Compute Cost	Local Context	Global Context
Full Attention	$\mathcal{O}(N^2)$	Lossless	Lossless
Sliding Window Attn.	$\mathcal{O}(N)$	Lossless	Limited
TTT / Linear Attn.	$\mathcal{O}(N)$	Compressed	Compressed
Ours (Hybrid Memory)	$\mathcal{O}(N)$	Lossless	Compressed

These failures suggest that a single memory strategy is fundamentally insufficient. To bridge this gap, we propose a learning-based *hybrid memory module* that maintains multi-scale geometric coherence within a fixed compute budget. As shown in Fig. 1, we formalize this via a dual-component memory system: a *non-parametric* sliding window attention (SWA) mechanism that preserves uncompressed local features and high-fidelity detail for the most recent chunks, and a *parametric* associative memory (Test-Time Training (Sun et al., 2024)) that compresses global context. This hybrid design effectively decouples these tasks: the long-range parametric memory anchors the global coordinate frame to prevent scale drift, while the short-range non-parametric memory ensures seamless, high-precision transitions.

Furthermore, to evaluate these mechanisms at an unprecedented scale, we introduce a long-context reconstruction benchmark derived from the VBR dataset (Brizi et al., 2024), spanning up to 19k frames and 11.5 km in trajectory. Remarkably, our architectural design enables the model to be trained on sequences of 128 frames, generalize seamlessly up to 1k frames during inference, and scale effectively up to 19k frames with periodic state resets and an optional feedforward pose-alignment. As shown in Fig. 1, LoGeR effectively breaks both the context and data walls. It substantially outperforms prior state-of-the-art feedforward methods on KITTI, reducing the Absolute Trajectory Error (ATE) from **72.86 to 18.65**, and achieves a **30.8%** relative improvement on the extremely long sequences of our VBR benchmark.

2. Related Work

Learning-based visual SLAM. Recent learning-based visual SLAM methods demonstrate outperforming classical approaches (Mur-Artal et al., 2015; Mur-Artal & Tardós, 2017; Engel et al., 2014; Newcombe et al., 2011) by learning strong 3D priors from training datasets (Teed & Deng, 2021; Lipson et al., 2024; Li et al., 2025) or leveraging powerful pretrained visual geometry models (Deng et al., 2025; Maggio et al., 2025). Yet these methods retain (relatively) expensive backends for graph construction, loop closure, and global optimization. While slower than feedforward methods, such SLAM-based pipelines represent the only viable approach for maintaining coherence over long contexts. Here, we provide a fully feedforward alternative that outperforms strong SLAM systems in our long-context evaluation, without relying on any backend optimization.

Feedforward 3D reconstruction. Recent work employs feedforward models for 3D reconstruction, directly outputting pointmaps in a canonical space (Wang et al., 2024; Leroy et al., 2024; Zhang et al., 2025a; Wang et al., 2025a; 2026). Despite achieving high-quality results, these methods can only process a limited number of input frames due to high memory cost from global attention layers, hindering real-world applications (*e.g.* robotics or autonomous driving) that require global consistency over long streams.

Long sequence reconstruction. Recent feedforward approaches tackle long sequence reconstruction using external spatial memory (Chen et al., 2025; Wang & Agapito, 2025; Wu et al., 2025; Lan et al., 2025), RNN-style persistent state (Wang et al., 2025b), or efficiency-oriented VGGT variants with sparse (Shen et al., 2026; Yuan et al., 2026) or causal (Zhuo et al., 2026) attention. However, their lack of demonstrated results on exceptionally long sequences (*e.g.*, minute-long) leaves their scalability in sequence lengths unverified. Concurrently, TTT3R (Chen et al., 2026) introduces a confidence-based update for single-frame streaming (Wang et al., 2025b). However, this frame-wise linear approach lacks the expressivity to capture complex temporal context or leverage the powerful multi-frame reasoning of bidirectional backbones (Wang et al., 2025a; 2026).

To overcome this, LoGeR processes streams in a chunk-wise manner. Unlike frame-wise updates, our design uniquely preserves the high-fidelity multi-frame reasoning power of bidirectional backbones while leveraging a hybrid memory mechanism to propagate information across chunks.

Memory for long-context modeling. Efficient long-sequence architectures revisit recurrent/state-space models (Gu et al., 2022; Gu & Dao, 2024) and linear attention (Katharopoulos et al., 2020; Schlag et al., 2021) to reduce the quadratic cost of Transformers. A complementary line of work uses local attention (*e.g.*, sliding-window attention) (Beltagy et al., 2020; Zaheer et al., 2020) to preserve high-fidelity short-range interactions, but such designs have limited access to global context. Recently, fast-weight mechanisms such as Test-Time Training (TTT) (Sun et al., 2024; Zhang et al., 2025b) treat memory as an evolving parameter state, enabling compact accumulation of long-range information, albeit with an inherent compression trade-off. As summarized in Tab. 1, these individual mechanisms inherently trade off computational cost, local context preservation, and global context accessibility.

While hybrid architectures in language models frequently combine these efficient mechanisms with full global attention to maintain performance (*e.g.*, Longformer (Beltagy et al., 2020) or Jamba (Lenz et al., 2025)), the extreme token density in dense vision prediction tasks makes this computationally prohibitive. Our method therefore introduces a hybrid memory that remains linear in sequence length, syner-

gizing non-parametric SWA for precise adjacent alignment with parametric TTT for long-range global consistency.

Dataset. Existing training datasets include both real (ARKitScenes, DL3DV (Ling et al., 2024), MegaDepth (Li & Snavely, 2018), ScanNet (Dai et al., 2017), Waymo (Schwall et al., 2020)) and synthetic ones (HyperSim (Roberts et al., 2021), Spring (Mehl et al., 2023), TartanAir (Wang et al., 2020), TartanAirV2 (Patel et al., 2025), UnReal4K (Aleotti et al., 2021), Virtual KITTI 2 (Cabon et al., 2020), OmniWorld-Game (Zhou et al., 2025)). ScanNet and TUM are popular evaluation benchmarks. However, these datasets are fundamentally limited in scale—both temporally and spatially. Even though some sequences may contain up to a thousand frames, they typically capture spatially bounded, room-scale environments. To truly evaluate long-context geometric reconstruction in expansive spaces, we adapt the VBR dataset for our evaluation. Spanning from 8,815 to 18,846 frames and covering extensive spatial trajectories up to 11.5 km, this benchmark presents a significant challenge for existing methods. Our chunk-wise formulation allows models trained on these limited datasets to generalize effectively to the expansive VBR benchmark.

3. Preliminary

We first provide the technical background for our approach.

3D reconstruction models. Existing methods often employ bidirectional transformer architectures to capture complex geometric information. Given a set of N multi-view images or video frames as input $\mathbf{I} = \{I_i\}_{i=1}^N$, where $I_i \in \mathbb{R}^{H \times W \times 3}$, the transformer predicts geometry as local pointmaps $\mathbf{P}_i \in \mathbb{R}^{H \times W \times 3}$ in *local camera coordinates*, alongside camera poses $\mathbf{c}_i \in \mathbb{R}^{4 \times 4}$ in *global world coordinates*. While effective for short context, the transformer-based methods are limited to bounded scenes.

Test-Time Training (TTT). TTT (Sun et al., 2024) introduces an architectural component designed to store useful context from prior forward calls of the model. It achieves this using *fast weights*, a set of parameters updated during both train and inference time. This contrasts with the model’s base parameters, or *slow weights*, which remain frozen during inference.

Let $\mathbf{x} = [x_1, x_2, \dots, x_N]$ be a 1D sequence of tokens where each token $x_i \in \mathbb{R}^d$. Following standard attention mechanism, each input token x_i is then projected into queries q_i , keys k_i , and values v_i , with $q_i, k_i, v_i \in \mathbb{R}^d$. Formally, TTT defines a neural network $f_W(\cdot) : \mathbb{R}^d \rightarrow \mathbb{R}^d$ parameterized by the fast weights W , which are then utilized in two operations.

$$\text{Update operation: } W \leftarrow W - \eta \nabla_W \mathcal{L}(f_W(\mathbf{k}), \mathbf{v}) \quad (1)$$

where η is the learning rate and $\mathcal{L}(\cdot, \cdot)$ is a loss function be-

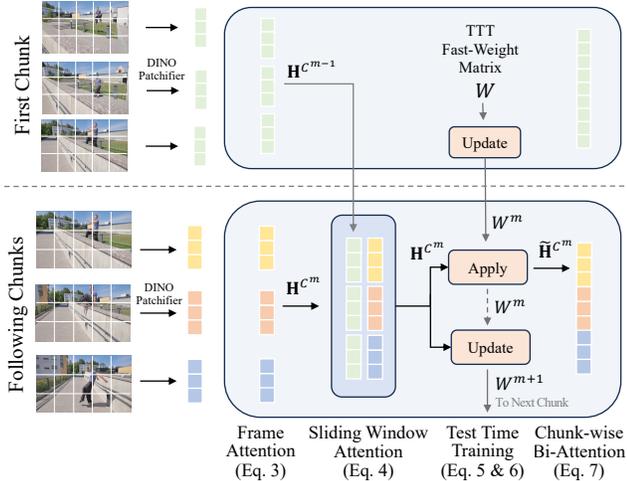


Figure 2. **Overview of a single block of our hybrid memory module.** We process the input sequence in consecutive chunks of frames. While each block utilizes frame and bidirectional attention from prior work, we introduce new components to effectively propagate information across the entire sequence. Specifically, we incorporate Sliding Window Attention to improve consistency between neighboring chunks, and Test-Time Training layers to maintain long-range, global consistency across all chunks.

tween the transformed key $f_W(k)$ and the value v . This loss encourages the function f_W to link keys with corresponding values, conceptually trying to encode the KV cache into the W matrix as a form of neural memory.

$$\text{Apply operation: } o = f_W(q), \quad (2)$$

where the updated fast weights W process the query q to compute the output vector o . The per-token TTT layer iteratively performs the update and applies operations on each token x_i in sequence. Following previous literature, we implement the fast weight architecture using SwiGLU layer and employ the Muon optimizer (Jordan et al., 2024) for the test-time updates. While TTT can compress and propagate information for long contexts with linear complexity, its compressed memory is inherently lossy. Next, we will introduce our hybrid-memory approach that uses different memory mechanisms to handle geometric information at various time scales over a long context.

4. Approach

4.1. Long-Context 3D Reconstruction Architecture

Motivation. To scale feedforward dense 3D reconstruction to minutes-long videos, we must overcome the quadratic complexity of global attention and the scarcity of long-horizon training data. End-to-end *chunk-wise processing* emerges as the natural solution: it tightly bounds the computational cost and ensures that local inferences remain within the distribution of existing short-context training data. However, independently processing chunks inherently breaks

global consistency. Therefore, we need a feedforward architecture that simultaneously provides: (i) strong local bidirectional reasoning within a short chunk to preserve dense geometric fidelity, (ii) a lossless short-range information highway to preserve high-precision geometric alignment across adjacent chunk boundaries, and (iii) a linear-time, fixed-size memory mechanism for long-range global propagation across thousands of frames.

Overview. We process input video streams sequentially by chunk, as shown in Figs. 1 and 2. Given a video $\mathcal{X} = \{I_t\}_{t=1}^T$, we partition it into M chunks, $\{\mathcal{C}^m\}_{m=1}^M$, with minimal overlap (e.g., a single overlapping frame). Each chunk \mathcal{C}^m comprises a potentially variable number of frames \mathcal{C}_j^m , where j indexes the frames within the chunk. Within each chunk, we employ a strong bidirectional geometry backbone (e.g., VGGT or π^3) to obtain high-quality dense predictions. To propagate information across chunks, we introduce two complementary mechanisms:

(1) *Long-term, lossy compression via chunk-wise TTT.* We insert TTT layers that maintain a set of fast weights W across a large number of chunks. Aligning with our chunk-wise processing, we utilize Large-Chunk Test-Time Training (LaCT) (Zhang et al., 2025b), which has been shown to be more efficient than standard TTT. During inference, the weights undergo both update and apply operations for each chunk. In the apply operation, the TTT layers use historical information stored in the weights to modulate the network’s processing of the current chunk. In the update operation, the weights are edited to store information from the current chunk, conceptually compressing important but redundant geometric information, e.g., coarse geometry and scale of the scene. While these fast weights theoretically offer an infinite receptive field, their practical capacity is inherently bounded by the training context length. To prevent drift over extremely long horizons, we optionally employ periodic state resets during inference (detailed in Sec. 5.1).

(2) *Short-term, lossless transfer via sliding-window attention (SWA).* Relying solely on TTT-style state passing is inherently lossy, which is problematic for dense 3D reconstruction where geometric consistency across adjacent frames is critical. To mitigate this, we *sparsely insert* sliding-window attention layers that attend to the tokens output by the frame attention layer from both the previous and current chunks, i.e., $\mathcal{C}^{m-1} \cup \mathcal{C}^m$. This establishes a lossless information highway that directly propagates high-fidelity features from the previous chunk. Importantly, this operation remains efficient with bounded compute and memory, as sliding window attention is applied only between neighboring chunks and inserted at a subset of network depths (only four layers).

These two cross-chunk pathways are complementary: TTT provides scalable long-range memory, while SWA ensures fine-grained geometric consistency across adjacent chunks.

Block structure. The geometry backbone first patchifies images into tokens and feeds them into a stack of residual network blocks. In each network block, we introduce our hybrid memory system as shown in Fig. 2. Denote slow weights (frozen at inference) by θ . Let \mathcal{C}^m be a chunk of frames C_1^m, \dots, C_n^m ; $\mathbf{H}^{C_i^m}$ be the token sequence specific to the frame C_i^m ; $\mathbf{H}^{\mathcal{C}^m}$ be the full token sequence entering a block for chunk m ; and $\text{LN}(\cdot)$ denote LayerNorm. We list the detailed structure of one block below:

(1) *Per-frame attention.* We apply self-attention independently to each frame’s tokens to extract spatial features:

$$\mathbf{H}^{C_i^m} \leftarrow \mathbf{H}^{C_i^m} + [\text{Attn}_{\text{frame}}(\text{LN}(\mathbf{H}^{C_i^m}); \theta), |i \in \{1, \dots, n\}] \quad (3)$$

where \square is the concatenation operator.

(2) *Sparse sliding-window attention (SWA) over \mathcal{C}^{m-1} and \mathcal{C}^m .* To align adjacent chunks, we insert SWA layers at a subset of depths (only 4 layers) to stay compute-bound:

$$\mathbf{H}^{\mathcal{C}^m} \leftarrow \mathbf{H}^{\mathcal{C}^m} + \text{Attn}_{\text{swa}}([\text{LN}(\mathbf{H}^{\mathcal{C}^{m-1}}), \text{LN}(\mathbf{H}^{\mathcal{C}^m})]; \theta). \quad (4)$$

(3) *Chunk-wise TTT layer (fast weights).* To integrate global context, we maintain a set of fast weights W^m that summarize information up to chunk m . The TTT layer performs an *apply-then-update* procedure at the chunk level. We use pre-norm inside TTT to stabilize long-horizon streaming.

$$\textbf{Apply: } \tilde{\mathbf{H}}^{\mathcal{C}^m} = \mathbf{H}^{\mathcal{C}^m} + f_{W^m}(\text{LN}(\mathbf{H}^{\mathcal{C}^m})), \quad (5)$$

$$\textbf{Update: } W^{m+1} = \mathcal{U}(W^m; \mathbf{H}^{\mathcal{C}^m}), \quad (6)$$

where $f_{W^m}(\cdot)$ is the fast-weight module (e.g., a SwiGLU MLP) parameterized by W^m , and $\mathcal{U}(\cdot)$ denotes the on-line update rule (e.g., a gradient-based update with a self-supervised objective). Intuitively, the *apply* step injects the current memory into token representations, while the *update* step writes the chunk’s summary into W for the next chunk.

(4) *Chunk-wise bidirectional attention within chunk \mathcal{C}^m .* Finally, within each chunk, we employ a bidirectional attention module for powerful geometric reasoning under a bounded context window. The updated representation $\tilde{\mathbf{H}}^{\mathcal{C}^m}$ then serves as the input to the subsequent network block:

$$\mathbf{H}^{\mathcal{C}^m} \leftarrow \tilde{\mathbf{H}}^{\mathcal{C}^m} + \text{BiAttn}_{\text{chunk}}(\text{LN}(\tilde{\mathbf{H}}^{\mathcal{C}^m}); \theta). \quad (7)$$

Prediction heads. After the final residual block, we attach lightweight decoders—a pointmap decoder and a camera-pose decoder—following π^3 , to produce the final dense pointmap predictions and per-frame camera poses.

4.2. Learning Objectives

Objective. We follow π^3 (Wang et al., 2026) and train LoGeR with (i) a *scale-invariant local pointmap loss* and

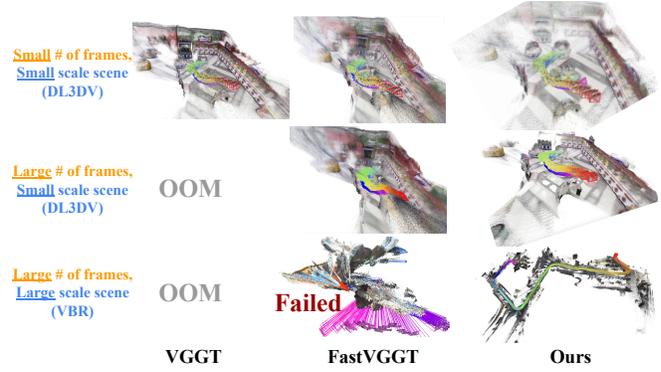


Figure 3. Comparison of different methods across varying sequence lengths and scene scales. Although FastVGGT is able to process a larger number of frames during inference, it fails completely on large-scale scenes, highlighting the inherent “data wall” of models trained strictly on short-context bubbles. In contrast, LoGeR breaks both the context and data walls by pairing a hybrid memory architecture with diverse long-horizon training data.

(ii) an *affine-invariant relative pose loss*, where the losses do not require a reference view. We align the predicted local pointmaps with a single per-sequence scale s^* as in MoGe (Wang et al., 2025c), and compute a reconstruction loss, normalized by depth values, over all pixels. To further over-constrain long-sequence training, we additionally impose a *global pointmap loss* on world-coordinate pointmaps obtained by transforming local pointmaps using the predicted camera poses.

$$\mathcal{L}_{\text{local}} = \frac{1}{N|\Omega|} \sum_{i=1}^N \sum_{p \in \Omega} \frac{1}{z_{i,p}} \|s^* \hat{\mathbf{x}}_{i,p} - \mathbf{x}_{i,p}\|_1, \quad (8)$$

$$\mathcal{L}_{\text{pose}} = \sum_{(i,j) \in \mathcal{P}} \left(\lambda_r \mathcal{L}_{\text{rot}}(\hat{\mathbf{R}}_{ij}, \mathbf{R}_{ij}) + \lambda_t \|s^* \hat{\mathbf{t}}_{ij} - \mathbf{t}_{ij}\|_{\text{Huber}} \right), \quad (9)$$

$$\mathcal{L}_{\text{global}} = \frac{1}{N|\Omega|} \sum_{i=1}^N \sum_{p \in \Omega} \|\Pi(\hat{\mathbf{T}}_i, \hat{\mathbf{x}}_{i,p}) - \Pi(\mathbf{T}_i, \mathbf{x}_{i,p})\|_1, \quad (10)$$

$$\mathcal{L} = \mathcal{L}_{\text{local}} + \mathcal{L}_{\text{pose}} + \lambda_{\text{global}} \mathcal{L}_{\text{global}}. \quad (11)$$

Here, N is the number of frames used in the supervision set, Ω is the pixel index set (so $|\Omega| = HW$), $\hat{\mathbf{x}}_{i,p} \in \mathbb{R}^3$ and $\mathbf{x}_{i,p} \in \mathbb{R}^3$ denote predicted and ground-truth local point coordinates at pixel p in frame i , and $z_{i,p}$ is the corresponding depth for normalization. $\hat{\mathbf{T}}_i = [\hat{\mathbf{R}}_i \mid \hat{\mathbf{t}}_i] \in \text{SE}(3)$ is the predicted camera pose, and $\hat{\mathbf{R}}_{ij}, \hat{\mathbf{t}}_{ij}$ denote the predicted relative motion between frames i and j (defined analogously for ground truth). \mathcal{P} is the set of supervised frame pairs (e.g., pairs within a chunk and/or overlap pairs across different chunks). $\Pi(\mathbf{T}, \mathbf{x})$ maps a local point \mathbf{x} to world coordinates using pose \mathbf{T} .

LoGeR* feedforward alignment. Despite having TTT and SWA, very long streams may still accumulate prediction

Table 2. Comparison of Absolute Trajectory Error (ATE \downarrow , m) on KITTI. The top and bottom blocks denote *optimization-based* and *feedforward* methods, respectively. **Bold** and underline indicate the best and second-best performance among feedforward methods. LoGeR* denotes a variant of our method with feedforward pose alignment for both training and inference.

Methods	00	01	02	03	04	05	06	07	08	09	10	Avg.
<i>num. of frames, scale</i>	4542, 3.7km	1101, 2.5km	4661, 5.1km	801, 0.6km	271, 0.4km	2761, 2.2km	1101, 1.2km	1101, 0.7km	4071, 3.2km	1591, 1.7km	1201, 0.9km	-
<i>Contains Loop?</i>	✓	×	✓	×	×	✓	✓	✓	×	✓	×	-
Opt.-based												
DROID-SLAM (Teed & Deng, 2021)	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 (Lipson et al., 2024)	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++ (Lipson et al., 2024)	8.30	11.86	39.64	2.50	0.78	5.74	11.60	1.52	110.90	76.70	13.70	25.75
VGGT-Long (Deng et al., 2025)	8.67	121.17	32.08	6.12	4.23	8.31	5.34	4.63	53.10	41.99	18.37	27.64
Feedforward												
FastVGGT (Shen et al., 2026)	OOM	639.39	OOM	21.53	9.51	OOM	40.56	51.35	OOM	201.54	196.22	-
InfiniteVGGT (Yuan et al., 2026)	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 (Wang et al., 2025b)	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 (Chen et al., 2026)	119.94	99.59	238.07	16.83	3.98	36.38	47.20	11.62	107.33	86.96	33.58	72.86
Pi3-Chunk (Proposed Baseline)	26.65	196.04	157.92	<u>5.13</u>	1.09	12.79	27.66	<u>5.94</u>	61.26	56.31	21.96	52.07
LoGeR (Ours)	62.34	41.64	<u>39.64</u>	4.89	<u>1.82</u>	41.27	<u>13.99</u>	16.24	<u>26.46</u>	<u>22.71</u>	8.84	<u>25.44</u>
LoGeR* (Ours)	<u>30.47</u>	<u>47.91</u>	36.32	5.38	1.95	<u>26.34</u>	6.60	5.55	24.41	10.12	<u>10.11</u>	18.65

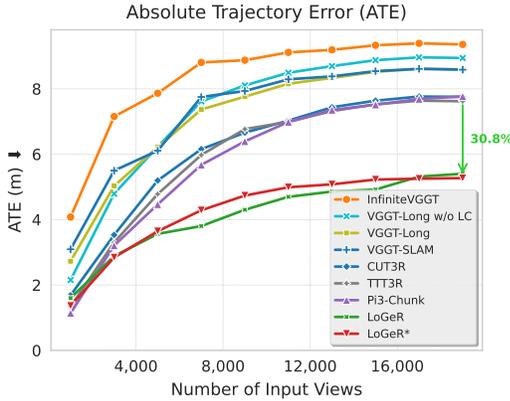


Figure 4. Quantitative results on our proposed VBR (Brizi et al., 2024) evaluation showing results on very long sequences spanning from 1,000 to 19,000 frames. Our methods achieve 30.8% more accurate results than prior methods.

errors. To mitigate this, we introduce LoGeR*, a variant that incorporates a purely feedforward alignment step to align raw predictions into a consistent global coordinate system. Let $\hat{\mathbf{T}}_k^{(m)}$ denote the raw predicted pose of the *overlapping frame* k within the current chunk \mathcal{C}^m , and let $\tilde{\mathbf{T}}^{(m-1)}$ denote the aligned poses of chunk \mathcal{C}^{m-1} (initialized as $\tilde{\mathbf{T}}^{(1)} = \hat{\mathbf{T}}^{(1)}$). We compute a rigid SE(3) alignment \mathbf{A}_m that maps the current chunk \mathcal{C}^m to the aligned coordinate system of the previous chunk \mathcal{C}^{m-1} using the overlap: $\mathbf{A}_m = \tilde{\mathbf{T}}_k^{(m-1)} (\hat{\mathbf{T}}_k^{(m)})^{-1}$. This transformation is applied to all frames in \mathcal{C}^m :

$$\tilde{\mathbf{T}}_t^{(m)} = \mathbf{A}_m \hat{\mathbf{T}}_t^{(m)}, \quad \forall t \in \mathcal{C}^m. \quad (12)$$

We use $\tilde{\mathbf{T}}_t^{(m)}$ as the final camera pose prediction of LoGeR* for *both* training and inference.

4.3. Data and Curriculum

The ‘Data Wall’. We posit that architectural improvements alone are insufficient for infinite-context reconstruction. As shown in Fig. 3, we observe that strong baselines like VGGT (Wang et al., 2025a), even when equipped with inference-time architectural efficiency improvement

(FastVGGT (Shen et al., 2026)), fail to generalize to large-scale scenes when trained solely on short-context or small-scene-scale data. To overcome this ‘data wall’, we construct a training mixture heavily weighted towards large-scale-scene datasets, *e.g.*, TartanAirV2 (Patel et al., 2025), which provides the necessary long-horizon signal for learning effective geometry compression.

Curriculum Training. Crucially, to stabilize the optimization of recurrent TTT layers, we employ a **progressive curriculum strategy**. By starting with easier sequences and progressively increasing complexity, we force the model to shift reliance from local Sliding Window Attention to the global TTT hidden state. Our schedule proceeds in three stages: (1) we begin with 48 frames split into 4 chunks; (2) we gradually increase chunk density to 12 chunks while maintaining fixed sequence length; and (3) leveraging H200 GPUs, we scale the context length to 128 frames, progressively increasing to 20 chunks. For LoGeR*, we initialize from the first-stage model, integrate the feedforward alignment, and fine-tune through the remaining curriculum. This strategy not only improves training efficiency by reducing train-time rollout overhead, but also boosts final performance, as in the ablation study in Secs. 5.3 and 6.

5. Experiments

Implementation details. We train our model on a mixture of real and synthetic datasets, including ARKitScenes (Baruch et al., 2021), DL3DV (Ling et al., 2024), HyperSim (Roberts et al., 2021), MegaDepth (Li & Snavely, 2018), ScanNet (Dai et al., 2017), ScanNet++ (Yeshwanth et al., 2023), Spring (Mehl et al., 2023), TartanAir (Wang et al., 2020), TartanAirV2 (Patel et al., 2025), UnReal4K (Wang et al., 2023), Virtual KITTI 2 (Cabon et al., 2020), Waymo (Schwall et al., 2020), and a subset of OmniWorld-Game. We optimize the model with AdamW (Loshchilov & Hutter, 2019) for 40k steps with a batch size of 32. The training process requires approximately two days on 32 NVIDIA H100 GPUs, followed by another two days on 32 H200 GPUs. We initialize the

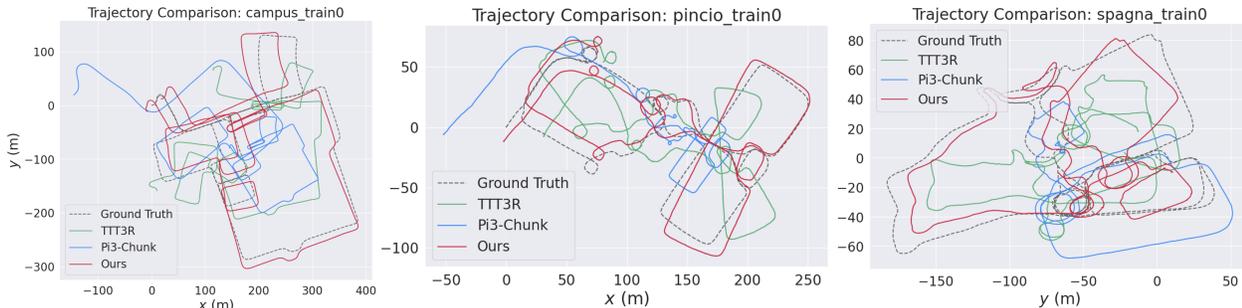


Figure 5. **Qualitative camera trajectories on the VBR dataset.** LoGeR accurately preserves global scale and trajectory over very long sequences, closely matching the ground truth where prior methods suffer from severe drift.

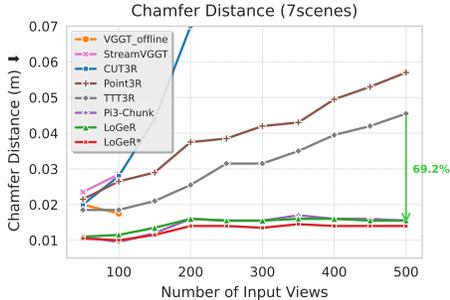


Figure 6. **3D reconstruction result on 7scenes.** Both our proposed baseline and LoGeR significantly outperform prior work by **69.2%**.



Figure 7. **Qualitative results on 3D reconstruction for the 7scenes dataset** demonstrating the LoGeR outperforms both our baseline and prior work TTT3R. LoGeR accurately reconstructs the bookshelf, while both our proposed strong baseline and TTT3R cause large distortions.

weights of the patchifier, frame attention, and chunk-wise bidirectional attention modules from π^3 (Wang et al., 2026). All the evaluation is done on a NVIDIA A100 40GB. See more details in Appendix Secs. A.1 to A.3.

5.1. Evaluation on Long Sequences

Benchmarks. As shown in Fig. 3, for large-scale geometric reconstruction, evaluation must be conducted on expansive scenes. To this end, we evaluate our method on two long-sequence benchmarks. First, we use the standard **KITTI benchmark** (Geiger et al., 2012), which contains sequences up to 4,661 frames and trajectory lengths up to 5 km. To further challenge the robustness of our method on even more complex and longer trajectories, we identify and repurpose the **VBR dataset** (Brizi et al., 2024). This dataset consists of video sequences taken from Rome with paired ground truth 3D reconstructions derived from refined LiDAR point clouds and bundle-adjusted camera poses. Specifically, we use 7 sequences ranging from 8,815 to 18,846 frames and covering distances between 1.4 km and 11.5 km. We report the Absolute Trajectory Error (ATE) of the predicted trajectory with respect to the ground truth after Umeyama alignment (Umeyama, 1991), following standard protocols (Teed & Deng, 2021; Lipson et al., 2024).

Baselines. We conduct a comprehensive comparison against both optimization-based and feedforward approaches. For *optimization-based* methods, we compare with classic dense SLAM systems such as DROID-SLAM (Teed & Deng,

2021) and DPV-SLAM (Lipson et al., 2024), as well as recent foundation-model-based approaches like VGGT-SLAM and VGGT-Long (Wang et al., 2025a). For *feed-forward* methods, we evaluate recurrent architectures including CUT3R (Wang et al., 2025b) and TTT3R (Chen et al., 2026), alongside bidirectional models such as FastVGGT (Shen et al., 2026) and autoregressive approaches like InfiniteVGGT (Yuan et al., 2026). For CUT3R and TTT3R, we adopt the reset algorithm as proposed in TTT3R to obtain a reasonable result. For our method, we also reset the fast weights in the TTT layers after every five windows to avoid error accumulation within a fixed size of state (Ruiz & Gu, 2025). Note, we also apply the feedforward pose alignment when doing a reset. See Appendix Sec. A.4 for more details. Based on the philosophy of chunk-causal, we also propose a simple baseline built on top of π^3 for long sequence reconstruction. Specifically, we process the input in a chunk-wise manner as in LoGeR, and then compute a SIM(3) transformation based on the overlapping frames to stitch the predictions of different chunks together. We name this new baseline *Pi3-Chunk*. See more details in Appendix Sec. A.5.

Results. Quantitatively, both LoGeR and our proposed baseline, Pi3-Chunk, significantly outperform existing feedforward methods on the **KITTI benchmark** (Tab. 2). Notably, LoGeR achieves an average performance that surpasses even the strongest optimization-based method, VGGT-Long, by 32.5%. This advantage is particularly evident on open-loop trajectories (*i.e.*, 01, 03, 04, 08, and 10), where our

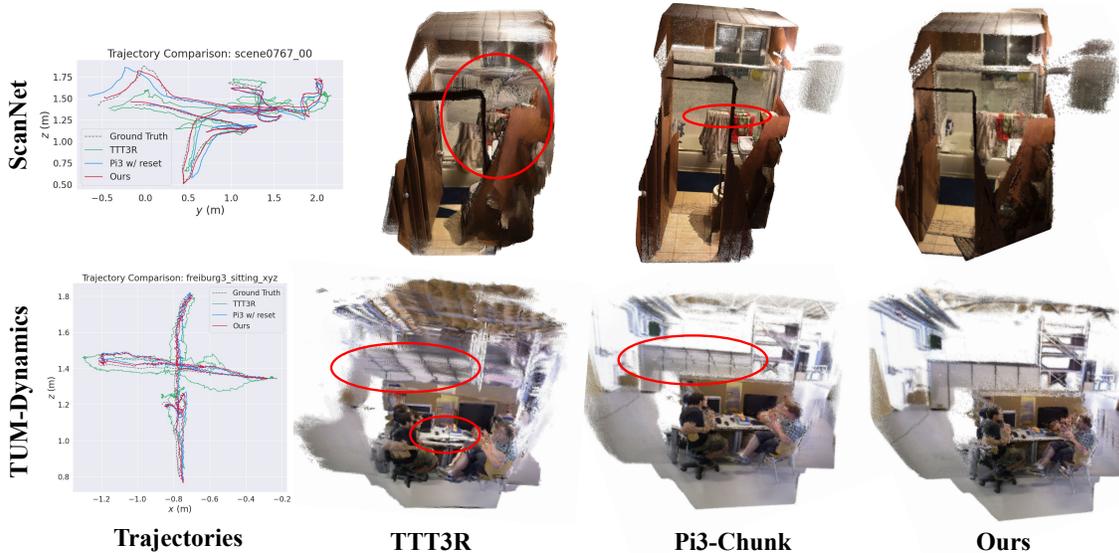


Figure 8. **Qualitative results on ScanNet and TUM-Dynamics.** While the proposed Pi3-Chunk baseline yields slightly better pose metrics on small-scale TUM sequences (unlike its severe drift on longer trajectories), LoGeR produces visually superior reconstructions. As highlighted, LoGeR accurately recovers structural details, avoiding the severe distortions and geometric artifacts present in baselines.

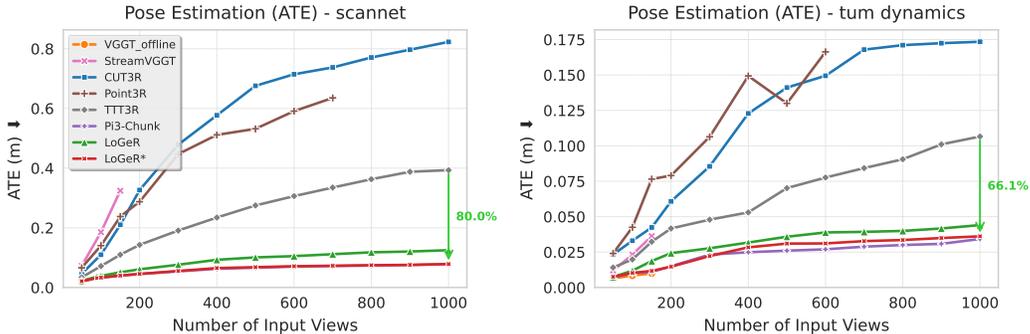


Figure 9. **Quantitative results on camera pose evaluation,** demonstrating that both our proposed baseline and LoGeR substantially outperform prior work, with **80.0%** and **66.1%** relative gains on ScanNet and TUM datasets, respectively.

method effectively mitigates accumulated drift without relying on loop closure. On the **VBR benchmark**, consistent improvements are observed quantitatively in Fig. 4 and qualitatively in Fig. 5. While Pi3-Chunk yields slightly better results at shorter horizons (*e.g.*, 1k frames), the advantage of LoGeR becomes increasingly prominent as the sequence length grows. This is because Pi3-Chunk relies on local overlapping frames for SIM(3) scale estimation, causing scale errors to accumulate exponentially over extended distances. In contrast, LoGeR’s TTT module inherently anchors the global scale. This is further evidenced by qualitative visualizations, where LoGeR maintains global scale consistency up to 20k frames, whereas the baseline exhibits severe scale drift in such extremely long sequences. See Appendix Secs. B.3 to B.5 for more long sequence results.

5.2. Evaluation on Short Sequences

Following TTT3R, we extend our evaluation to shorter sequences (up to 1k frames). First, we evaluate 3D point-cloud reconstruction on 7-Scenes (Shotton et al., 2013) with

sequence lengths ranging from 50 to 500 frames. Subsequently, we evaluate camera pose estimation on ScanNetV2 (Dai et al., 2017) and TUM-Dynamics (Sturm et al., 2012) for sequence lengths ranging from 50 to 1k frames. For all experiments in this section, we use a chunk size of 64 and an overlap of 3 frames. We compare our approach with other learning-based sub-quadratic baselines, including explicit-state methods such as Point3R and implicit state-space models such as CUT3R, TTT3R, and StreamVGGT. Additionally, we include comparisons with bidirectional baselines, specifically VGGT and π^3 . Results for 7-Scenes are presented in Figs. 6 and 7, while results for ScanNetV2 and TUM-Dynamics are shown in Figs. 8 and 9. Across all comparisons, both our proposed baseline and LoGeR significantly outperform prior work. See Appendix Secs. B.1 and B.2 for more evaluation on short sequences.

5.3. Ablation Study

In this section, we validate the effectiveness of our proposed hybrid architecture, data mixture, and training curriculum.



Figure 10. **Ablation of disabling SWA or TTT at inference time.** SWA ensures geometric consistency between adjacent chunks (local consistency), and lacking it causes noticeable local misalignment artifacts. Meanwhile, TTT maintains the global context (global consistency); without it, the model suffers from severe trajectory drift over long horizons.

For computational efficiency, all ablation models are trained with a reduced number of frames compared to the final model and evaluated on a subset of ScanNet and the TUM-Dynamics dataset. Quantitative results are in Tab. 3.

Architecture Design. First, we assess the contribution of the two core memory components: the TTT layer and the SWA layer. As shown in the first block of Tab. 3, removing either component leads to a noticeable performance drop in ATE, confirming that the hybrid design is essential for robust state estimation. To further intuit the role of each module, we provide a qualitative comparison in Fig. 10, where we selectively disable the SWA or TTT layers of a trained LoGeR model at *inference time*. Visualizing the trajectory demonstrates that both components are critical: TTT ensures global consistency, while SWA maintains local smoothness, and lacking either results in significant drift or geometric degradation.

Dataset Mixture. Next, we verify the necessity of including large-scale navigation datasets to overcome the “data wall”. We train a variant of our model by excluding five specific large-scale datasets: TartanAir, TartanAirV2, Waymo, Virtual KITTI 2, and OmniWorld-Game. The results in the second block of Tab. 3 show that the model trained on the full data mixture significantly outperforms the one trained without these large-scale sources, validating our hypothesis that diverse, long-horizon data is key to generalization.

Curriculum Training. Finally, we analyze the impact of our progressive curriculum strategy. We compare the standard training schedule against our curriculum approach for both LoGeR and LoGeR*. As reported in the bottom blocks of Tab. 3, the curriculum strategy consistently improves performance across both model variants and datasets, demonstrating its effectiveness in stabilizing the optimization of recurrent layers and feedforward alignment.

6. Conclusion

We addressed the limitations of long-sequence estimation for feedforward 3D geometry models by introducing LoGeR, a complementary hybrid architecture combining Sliding Window Attention (SWA) with Test-Time Training (TTT) layer. Validated on our benchmark, LoGeR processes thou-

Table 3. **Comparison of baseline and ablation results** across ScanNet (subset) and TUM datasets. We report the results of ATE (\downarrow) on both 500 and 1000 frames of sequences.

Method	ScanNet		TUM	
	500f	1000f	500f	1000f
LoGeR	0.087	0.107	0.033	0.050
w/o TTT	0.108	0.162	0.043	0.079
w/o SWA	0.115	0.143	0.039	0.053
All datasets	0.087	0.107	0.033	0.050
w/o 5 large datasets	0.102	0.156	0.050	0.072
LoGeR	0.087	0.107	0.033	0.050
w/o curriculum	0.098	0.133	0.049	0.062
LoGeR*	0.070	0.080	0.031	0.036
w/o curriculum	0.078	0.093	0.029	0.040

sands of frames without optimization, outperforming other methods on long-sequence reconstruction with various representations, *e.g.*, explicit memories, recurrent states, or causal attention. Our work opens new avenues for long-context spatio-temporal reasoning in dynamic scenes, video understanding, and robotics.

Discussion and Future work

Our findings open several promising future directions. First, while TTT fast weights have a fixed memory footprint that theoretically allows infinite context, in practice they struggle to generalize beyond the number of chunks they were trained with (Ruiz & Gu, 2025), restricting their effective range to the training context length (which is constrained by hardware memory budgets). Exceeding this on extremely long sequences (*e.g.* $> 1,000$ frames) causes error accumulation and trajectory drift. Preventing this currently requires periodic state resets that sacrifice long-term context. We hope future linear sequence models will resolve this length-generalization bottleneck. Second, the availability of expansive, high-quality training data remains a significant bottleneck, described as the “data wall” discussed in Sec. 4.3. We hope future community efforts will focus on curating more diverse, long-horizon datasets. Finally, we aim to extend our hybrid memory architecture (SWA + TTT) beyond geometric reconstruction, exploring its potential in other domains that demand both long-term global consistency and strong local dependencies.

Acknowledgment

We would especially like to thank Noah Snavely for helpful feedback throughout the project. We thank Tianyuan Zhang, Songlin Yang, Youming Deng, Haiwen Feng, Qianqian Wang, and Yifei Zhang for helpful discussions. We thank Alfred Piccioni for help with the training infrastructure. We thank Xingyu Chen for providing details on the evaluation. We thank Angjoo Kanazawa, Tyler Bonnen, and Jiahui Lei for helpful feedback on the manuscript.

Impact Statement

This paper presents work whose goal is to address the problem of long-context 3D reconstruction design using a new architectural component. The main societal consequences of our work come from any potential improvements in the ability to convert video into dense 3D scenes, such as improved applications in VR, generative 3D, and Robotics. None of these consequences is specific to our work and applies to all techniques that improve the task of 3D reconstruction or design new architectural components for deep learning architectures.

References

- Aleotti, F., Tosi, F., Ramirez, P. Z., Poggi, M., Salti, S., Mattoccia, S., and Di Stefano, L. Neural disparity refinement for arbitrary resolution stereo. In *3DV*, pp. 207–217. IEEE, 2021.
- Baruch, G., Chen, Z., Dehghan, A., Dimry, T., Feigin, Y., Fu, P., Gebauer, T., Joffe, B., Kurz, D., Schwartz, A., et al. ARKitScenes: A diverse real-world dataset for 3D indoor scene understanding using mobile RGB-D data. *arXiv preprint arXiv:2111.08897*, 2021.
- Beltagy, I., Peters, M. E., and Cohan, A. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020.
- Brizi, L., Giacomini, E., Di Giammarino, L., Ferrari, S., Salem, O., De Rebotto, L., and Grisetti, G. VBR: A vision benchmark in Rome. In *ICRA*, pp. 15868–15874. IEEE, 2024.
- Cabon, Y., Murray, N., and Humenberger, M. Virtual KITTI 2. *arXiv preprint arXiv:2001.10773*, 2020.
- Chen, X., Chen, Y., Xiu, Y., Geiger, A., and Chen, A. TTT3R: 3D reconstruction as test-time training. In *ICLR*, 2026.
- Chen, Z., Qin, M., Yuan, T., Liu, Z., and Zhao, H. Long3R: Long sequence streaming 3D reconstruction. In *ICCV*, pp. 5273–5284, 2025.
- Dai, A., Chang, A. X., Savva, M., Halber, M., Funkhouser, T., and Nießner, M. ScanNet: Richly-annotated 3D reconstructions of indoor scenes. In *CVPR*, pp. 5828–5839, 2017.
- Deng, K., Ti, Z., Xu, J., Yang, J., and Xie, J. VGGT-Long: Chunk it, loop it, align it—pushing VGGT’s limits on kilometer-scale long RGB sequences. *arXiv preprint arXiv:2507.16443*, 2025.
- Dong, J., Feng, B., Guessous, D., Liang, Y., and He, H. Flex attention: A programming model for generating optimized attention kernels. *arXiv preprint arXiv:2412.05496*, 2(3):4, 2024.
- Elflein, S., Li, R., Agostinho, S., Gojcic, Z., Leal-Taixé, L., Zhou, Q., and Osep, A. Vgg-t³: Offline feed-forward 3d reconstruction at scale. In *CVPR*, 2026.
- Engel, J., Schöps, T., and Cremers, D. LSD-SLAM: Large-scale direct monocular SLAM. In *ECCV*, pp. 834–849, 2014.
- Geiger, A., Lenz, P., and Urtasun, R. Are we ready for autonomous driving? the KITTI vision benchmark suite. In *CVPR*, pp. 3354–3361. IEEE, 2012.
- Gu, A. and Dao, T. Mamba: Linear-time sequence modeling with selective state spaces. In *COLM*, 2024.
- Gu, A., Goel, K., and Re, C. Efficiently modeling long sequences with structured state spaces. In *ICLR*, 2022.
- Jordan, K., Jin, Y., Boza, V., Jiacheng, Y., Cesista, F., Newhouse, L., and Bernstein, J. Muon: An optimizer for hidden layers in neural networks, 2024.
- Katharopoulos, A., Vyas, A., Pappas, N., and Fleuret, F. Transformers are RNNs: Fast autoregressive transformers with linear attention. In *ICML*, pp. 5156–5165. PMLR, 2020.
- Lan, Y., Luo, Y., Hong, F., Zhou, S., Chen, H., Lyu, Z., Yang, S., Dai, B., Loy, C. C., and Pan, X. Stream3R: Scalable sequential 3D reconstruction with causal transformer. *arXiv preprint arXiv:2508.10893*, 2025.
- Lenz, B., Lieber, O., Arazi, A., Bergman, A., Manevich, A., Peleg, B., Aviram, B., Almagor, C., Fridman, C., Padnos, D., et al. Jamba: Hybrid transformer-mamba language models. In *ICLR*, 2025.
- Leroy, V., Cabon, Y., and Revaud, J. Grounding image matching in 3D with MAST3R. In *ECCV*, pp. 71–91, 2024.
- Li, Z. and Snavely, N. MegaDepth: Learning single-view depth prediction from internet photos. In *CVPR*, pp. 2041–2050, 2018.

- Li, Z., Tucker, R., Cole, F., Wang, Q., Jin, L., Ye, V., Kanazawa, A., Holynski, A., and Snavely, N. MegaSaM: Accurate, fast and robust structure and motion from casual dynamic videos. In *CVPR*, pp. 10486–10496, 2025.
- Ling, L., Sheng, Y., Tu, Z., Zhao, W., Xin, C., Wan, K., Yu, L., Guo, Q., Yu, Z., Lu, Y., et al. DL3DV-10K: A large-scale scene dataset for deep learning-based 3D vision. In *CVPR*, pp. 22160–22169, 2024.
- Lipson, L., Teed, Z., and Deng, J. Deep patch visual SLAM. In *ECCV*, pp. 424–440. Springer, 2024.
- Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. In *ICLR*, 2019.
- Maggio, D., Lim, H., and Carlone, L. VGGT-SLAM: Dense RGB SLAM optimized on the SL (4) manifold. *arXiv preprint arXiv:2505.12549*, 2025.
- Mehl, L., Schmalfluss, J., Jahedi, A., Nalivayko, Y., and Bruhn, A. Spring: A high-resolution high-detail dataset and benchmark for scene flow, optical flow and stereo. In *CVPR*, pp. 4981–4991, 2023.
- Mur-Artal, R. and Tardós, J. D. ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras. *IEEE Transactions on Robotics*, 33(5):1255–1262, 2017.
- Mur-Artal, R., Montiel, J. M. M., and Tardos, J. D. ORB-SLAM: a versatile and accurate monocular SLAM system. *IEEE Transactions on Robotics*, 31(5):1147–1163, 2015.
- Newcombe, R. A., Lovegrove, S. J., and Davison, A. J. DTAM: Dense tracking and mapping in real-time. In *ICCV*, pp. 2320–2327, 2011.
- Palazzolo, E., Behley, J., Lottes, P., Giguere, P., and Stachniss, C. Refusion: 3d reconstruction in dynamic environments for RGB-D cameras exploiting residuals. In *IROS*, pp. 7855–7862, 2019.
- Patel, M., Yang, F., Qiu, Y., Cadena, C., Scherer, S., Hutter, M., and Wang, W. TartanGround: A large-scale dataset for ground robot perception and navigation. In *IROS*, 2025.
- Roberts, M., Ramapuram, J., Ranjan, A., Kumar, A., Bautista, M. A., Paczan, N., Webb, R., and Susskind, J. M. Hypersim: A photorealistic synthetic dataset for holistic indoor scene understanding. In *ICCV*, pp. 10912–10922, 2021.
- Ruiz, R. B. and Gu, A. Understanding and improving length generalization in recurrent models. In *ICML*, 2025.
- Schlag, I., Irie, K., and Schmidhuber, J. Linear transformers are secretly fast weight programmers. In *ICML*, pp. 9355–9366. PMLR, 2021.
- Schwall, M., Daniel, T., Victor, T., Favaro, F., and Hohnhold, H. Waymo public road safety performance data. *arXiv preprint arXiv:2011.00038*, 2020.
- Shen, Y., Zhang, Z., Qu, Y., and Cao, L. FastVGGT: Training-free acceleration of visual geometry transformer. In *ICLR*, 2026.
- Shotton, J., Glocker, B., Zach, C., Izadi, S., Criminisi, A., and Fitzgibbon, A. Scene coordinate regression forests for camera relocalization in RGB-D images. In *CVPR*, pp. 2930–2937, 2013.
- Sturm, J., Engelhard, N., Endres, F., Burgard, W., and Cremers, D. A benchmark for the evaluation of RGB-D SLAM systems. In *IROS*, pp. 573–580, 2012.
- Sun, Y., Li, X., Dalal, K., Xu, J., Vikram, A., Zhang, G., Dubois, Y., Chen, X., Wang, X., Koyejo, S., et al. Learning to (learn at test time): RNNs with expressive hidden states. *arXiv preprint arXiv:2407.04620*, 2024.
- Teed, Z. and Deng, J. DROID-SLAM: Deep visual SLAM for monocular, stereo, and RGB-D cameras. *NeurIPS*, pp. 16558–16569, 2021.
- Umeyama, S. Least-squares estimation of transformation parameters between two point patterns. *IEEE TPAMI*, 13(4):376–380, 1991.
- Wang, H. and Agapito, L. 3D reconstruction with spatial memory. In *3DV*, 2025.
- Wang, J., Chen, M., Karaev, N., Vedaldi, A., Rupprecht, C., and Novotny, D. VGGT: Visual geometry grounded transformer. In *CVPR*, pp. 5294–5306, 2025a.
- Wang, Q., Zhang, Y., Holynski, A., Efros, A. A., and Kanazawa, A. Continuous 3D perception model with persistent state. In *CVPR*, pp. 10510–10522, 2025b.
- Wang, R., Xu, S., Dai, C., Xiang, J., Deng, Y., Tong, X., and Yang, J. MoGe: Unlocking accurate monocular geometry estimation for open-domain images with optimal training supervision. In *CVPR*, pp. 5261–5271, 2025c.
- Wang, S., Leroy, V., Cabon, Y., Chidlovskii, B., and Revaud, J. DUST3R: Geometric 3D vision made easy. In *CVPR*, pp. 20697–20709, 2024.
- Wang, W., Zhu, D., Wang, X., Hu, Y., Qiu, Y., Wang, C., Hu, Y., Kapoor, A., and Scherer, S. TartanAir: A dataset to push the limits of visual SLAM. In *IROS*, pp. 4909–4916, 2020.

- Wang, Y., Shi, M., Li, J., Huang, Z., Cao, Z., Zhang, J., Xian, K., and Lin, G. Neural video depth stabilizer. In *ICCV*, pp. 9466–9476, 2023.
- Wang, Y., Zhou, J., Zhu, H., Chang, W., Zhou, Y., Li, Z., Chen, J., Pang, J., Shen, C., and He, T. π^3 : Permutation-equivariant visual geometry learning. In *ICLR*, 2026.
- Wu, Y., Zheng, W., Zhou, J., and Lu, J. Point3R: Streaming 3D reconstruction with explicit spatial pointer memory. In *NeurIPS*, 2025.
- Yeshwanth, C., Liu, Y.-C., Nießner, M., and Dai, A. ScanNet++: A high-fidelity dataset of 3D indoor scenes. In *ICCV*, pp. 12–22, 2023.
- Yuan, S., Yang, Y., Yang, X., Zhang, X., Zhao, Z., Zhang, L., and Zhang, Z. InfiniteVGGT: Visual geometry grounded transformer for endless streams. *arXiv preprint arXiv:2601.02281*, 2026.
- Zaheer, M., Guruganesh, G., Dubey, K. A., Ainslie, J., Alberti, C., Ontanon, S., Pham, P., Ravula, A., Wang, Q., Yang, L., et al. Big bird: Transformers for longer sequences. *NeurIPS*, 33:17283–17297, 2020.
- Zhang, J., Herrmann, C., Hur, J., Jampani, V., Darrell, T., Cole, F., Sun, D., and Yang, M.-H. MonST3R: A simple approach for estimating geometry in the presence of motion. In *ICLR*, 2025a.
- Zhang, T., Bi, S., Hong, Y., Zhang, K., Luan, F., Yang, S., Sunkavalli, K., Freeman, W. T., and Tan, H. Test-time training done right. *arXiv preprint arXiv:2505.23884*, 2025b.
- Zhou, Y., Wang, Y., Zhou, J., Chang, W., Guo, H., Li, Z., Ma, K., Li, X., Wang, Y., Zhu, H., et al. OmniWorld: A multi-domain and multi-modal dataset for 4D world modeling. *arXiv preprint arXiv:2509.12201*, 2025.
- Zhuo, D., Zheng, W., Guo, J., Wu, Y., Zhou, J., and Lu, J. Streaming 4D visual geometry transformer. In *ICLR*, 2026.

Appendix

Content

A. Additional Implementation Details	13
A.1. Details of Training Data	13
A.2. More Architecture Details	13
A.3. More Training Details	13
A.4. More Inference Details	14
A.5. More Details on the Proposed Baseline	14
B. More Experimental Results	15
B.1. More Evaluation on 7-Scenes	15
B.2. Depth Evaluation	15
B.3. More Detailed Results on VBR	16
B.4. More Qualitative Results on Trajectory	16
B.5. More Qualitative Results on 3D Reconstruction	17

A. Additional Implementation Details

A.1. Details of Training Data

To train our model to effectively handle long-context geometric reconstruction, we utilize a diverse mixture of 14 large-scale datasets, encompassing both real-world and synthetic scenes across indoor, outdoor, and autonomous driving environments.

To standardize the inputs for our chunk-wise architecture, all datasets are processed into multi-view sequences consisting of 48 views (or 128 views for the training stage on H200 GPUs) with a uniform resolution of 504×280 . The sampling strategy follows CUT3R (Wang et al., 2025b). Furthermore, to ensure the quality of geometric supervision, we apply rigorous depth filtering across all datasets. Depending on the dataset characteristics (whether metric scale or not), we utilize either a maximum depth threshold (*e.g.*, 80.0 meters for ARKitScenes and ScanNet) or a percentile-based clipping strategy (*e.g.*, 90th or 98th percentile for DL3DV and TartanAir) to mask out noisy or invalid depth values.

As discussed in the main text, overcoming the “data wall” requires providing the model with sufficient long-horizon signals and diverse scene priors. To achieve this, we construct our training mixture by heavily weighting large-scale navigation datasets (*e.g.*, TartanAirV2 and VKITTI2) to encourage long-range geometric reasoning. While DL3DV does not feature extremely long spatial trajectories like the navigation datasets, we also assign it a correspondingly high sampling weight due to its exceptional real-world scene diversity, which is critical for model generalization. Conversely, we down-weight smaller or object-centric datasets. Additionally, we note that at the time of training, only a subset of 5,000 sequences from the OmniWorld-Game dataset was publicly released, which comprises the portion we utilized in our mixture. The complete list of datasets and their

Table 4. Training data mixture. We list the datasets used in our final training configuration along with their relative sampling percentages. Navigation and large-scale scene datasets are heavily weighted to encourage long-range geometric reconstruction.

Dataset	Percentage (%)
DL3DV (Ling et al., 2024)	17.89
TartanAirV2 (Patel et al., 2025)	17.89
OmniWorld-Game (subset) (Zhou et al., 2025)	17.89
ARKitScenes (Baruch et al., 2021)	10.44
TartanAir (Wang et al., 2020)	8.94
Waymo (Schwall et al., 2020)	6.71
ARKitScenes HighRes (Baruch et al., 2021)	4.18
MegaDepth (Li & Snavely, 2018)	4.18
ScanNet (Dai et al., 2017)	4.18
ScanNet++ (Yeshwanth et al., 2023)	3.13
Virtual KITTI 2 (Cabon et al., 2020)	2.24
HyperSim (Roberts et al., 2021)	2.08
Spring (Mehl et al., 2023)	0.22
UnReal4K (Wang et al., 2023)	0.04

relative sampling percentages is summarized in Tab. 4.

A.2. More Architecture Details

The base network architecture is similar to π^3 (Wang et al., 2026), comprising a DINO-based patchifier, 18 residual blocks, and dedicated prediction heads for the local pointmap, global camera pose, and confidence map.

We insert a TTT layer into each of the 18 residual blocks, while the SWA layers are sparsely integrated only at the 6th, 10th, 14th, and 18th blocks. These SWA layers are initialized from the corresponding global attention layers in π^3 . Furthermore, to provide temporal context within the SWA layers, we add three learnable positional embedding tokens to the frame representations. These tokens explicitly indicate whether a frame overlaps with the previous chunk, has no overlap, or overlaps with the subsequent chunk. The TTT layer features a head dimension of 512 and an intermediate layer with an expansion factor of 4. In terms of model size, the base network contains approximately 954M parameters, while the introduced TTT and SWA layers account for an additional 296M parameters.

A.3. More Training Details

Optimization and Hyperparameters. We optimize our model using AdamW with momentum parameters $\beta = (0.9, 0.999)$. During training, we freeze the encoder and the prediction heads to retain their pre-trained feature representations. We apply decoupled learning rates: 5×10^{-4} for the newly introduced TTT and SWA layers, and a smaller learning rate of 1×10^{-5} for the unfrozen parameters of the base network. To prevent overfitting while maintaining training stability, we employ a selective weight decay strategy with a factor of 0.05. Specifically, weight decay is exclusively applied to multi-dimensional parameter tensors (*i.e.* ≥ 2 dimensions, such as weight matrices in linear or con-

volutional layers). One-dimensional or scalar parameters, including biases and normalization scale/shift parameters (e.g., in LayerNorm), remain undecayed. This practice prevents the disruption of normalization statistics and preserves network expressivity. For the training objective, the loss weights are empirically set to $\lambda_r = 0.1$ for rotation, $\lambda_t = 10$ for translation, and $\lambda_{\text{global}} = 1$ for the global pointmap.

Training Schedule and Curriculum. We use a cosine learning rate decay schedule with a warmup of 1,000 steps. To progressively build the model’s capacity for long-context reasoning, our curriculum training is split into two hardware-specific stages. In the first stage, we train the model for 25,000 steps on NVIDIA H100 GPUs using a fixed sequence length of 48 frames. Over this stage, we linearly decrease the chunk size (the number of frames per chunk) from 12 to 4, and the overlap size from 3 to 1, effectively increasing the number of recurrent steps. In the second stage, we leverage NVIDIA H200 GPUs to train for an additional 15,000 steps, extending the total sequence length to 128 frames. During this phase, we linearly decrease the chunk size from 12 to 8, and the overlap size from 3 to 2.

Efficiency Implementations. Given the extreme memory demands of unrolling long sequences and maintaining recurrent states, we heavily rely on gradient checkpointing across the network blocks to reduce memory consumption. Furthermore, we implement our chunk-wise sliding window attention (SWA) using FlexAttention (Dong et al., 2024) to optimize both memory footprint and compute efficiency during training.

A.4. More Inference Details

Inference Configurations. For the evaluation of short sequences and small-scale scenes, we employ a chunk size of 64 and an overlap size of 3 for both LoGeR and LoGeR*. For long sequences and large-scale environments, we maintain this exact configuration for LoGeR*. However, because the base LoGeR model is more susceptible to trajectory drift at larger scales, we reduce its chunk size to mitigate error accumulation. Specifically, for LoGeR, we use a chunk size of 32 for the KITTI dataset and 48 for the VBR dataset.

Latency and Efficiency. To optimize computational efficiency during inference, we implement a KV-cache mechanism for the block-wise SWA layers, avoiding redundant token computations. We evaluate the inference speed and peak memory consumption using a single NVIDIA A100 (40GB) GPU over a sequence of 500 frames. As detailed in Tab. 5, the performance scales predictably with the chosen chunk size, demonstrating a clear trade-off between the temporal context window and computational cost. We believe that inference efficiency can be further improved by pruning additional TTT layers or by employing strided sampling within the SWA module for historical, non-overlapping frames. We

Table 5. **Inference efficiency.** Measured on a single NVIDIA A100 (40GB) GPU over a 500-frame sequence with varying chunk sizes. As the number of frames increases, the speed improves slightly, while memory consumption remains constant.

Chunk Size	Speed (FPS)	Memory (GB)
64	9.3	27.2
48	10.6	22.3
32	12.1	18.1

leave these system-level optimizations for future work.

A.5. More Details on the Proposed Baseline

Based on the philosophy of chunk-causal processing, we introduce *Pi3-Chunk*, a simple baseline built on top of π^3 (Wang et al., 2026) for long sequence reconstruction. Specifically, we process the input sequence in a chunk-wise manner similar to LoGeR, and then rely on a feedforward alignment step to stitch the predictions of different chunks together using their overlapping frames.

While the alignment in LoGeR* strictly requires a rigid SE(3) transformation \mathbf{A}_m , *Pi3-Chunk* necessitates a SIM(3) transformation. This is because the base π^3 model predicts geometries and translations that are only consistent up-to-scale within each individual chunk. Therefore, to align the current chunk \mathcal{C}_m to the globally aligned coordinate system of the previous chunk \mathcal{C}_{m-1} , we must first recover and compensate for the relative scale shift.

Let $\hat{\mathbf{x}}_{k,p}^{(m)}$ denote the raw predicted local point coordinates of the overlapping frame k at pixel p in the current chunk \mathcal{C}_m , and let $\tilde{\mathbf{x}}_{k,p}^{(m-1)}$ denote the corresponding *scale-adjusted* local point coordinates from the previously aligned chunk \mathcal{C}_{m-1} . Because these pointmaps represent the same physical geometry in the camera coordinate system, their ratio directly reflects the scale ambiguity. We robustly estimate the relative scale factor s_m by taking the median (or alternatively, a truncated mean, which yields similar empirical performance) of the point-wise coordinate norms over all valid pixels $p \in \Omega$:

$$s_m = \text{Median}_{p \in \Omega} \left(\frac{\|\tilde{\mathbf{x}}_{k,p}^{(m-1)}\|_2}{\|\hat{\mathbf{x}}_{k,p}^{(m)}\|_2} \right). \quad (13)$$

Once the scale factor s_m is determined, we apply it to adjust the local pointmaps and the translation components of all raw poses in the current chunk. Let $\hat{\mathbf{T}}_t^{(m)} = [\hat{\mathbf{R}}_t^{(m)} \mid \hat{\mathbf{t}}_t^{(m)}]$ be the raw predicted pose. We obtain the scale-adjusted pose $\bar{\mathbf{T}}_t^{(m)}$ as:

$$\bar{\mathbf{T}}_t^{(m)} = [\hat{\mathbf{R}}_t^{(m)} \mid s_m \hat{\mathbf{t}}_t^{(m)}], \quad \forall t \in \mathcal{C}_m. \quad (14)$$

Additionally, the local pointmaps for this chunk are recursively scaled as $\tilde{\mathbf{x}}_{t,p}^{(m)} = s_m \hat{\mathbf{x}}_{t,p}^{(m)}$ to serve as the reference for the next chunk.

Table 6. Comparison of Absolute Trajectory Error (ATE \downarrow , m) on VBR. The top and bottom blocks denote optimization-based and feedforward methods, respectively. **Bold** and underline indicate the best and second-best performance among feedforward methods.

Methods		colosseo_0	campus_0	campus_1	pincio_0	spagna_0	diag_0	ciampino_1	Avg.
<i>num. of frames, scale</i>		8815, 1.45km	12042, 2.73km	11671, 2.95km	11142, 1.27km	14141, 1.56km	10021, 1.02km	18846, 5.20km	-
Opt.	VGGT-SLAM (Maggio et al., 2025)	10.05	9.67	8.47	8.15	7.55	5.80	11.14	8.69
	VGGT-Long w/o LC (Deng et al., 2025)	9.03	10.89	9.91	7.31	6.85	5.55	13.05	8.94
	VGGT-Long (Deng et al., 2025)	6.29	10.89	9.91	7.31	7.09	5.55	13.12	8.60
Feedforward	InfiniteVGGT (Yuan et al., 2026)	9.16	11.12	10.00	8.41	7.50	5.62	13.70	9.36
	CUT3R (Wang et al., 2025b)	9.09	6.50	6.57	6.83	6.68	5.35	13.26	7.75
	TTT3R (Chen et al., 2026)	8.69	7.71	7.52	5.82	6.11	4.30	13.18	7.62
	Pi3-Chunk (Proposed Baseline)	8.78	8.86	8.11	6.48	6.69	4.88	10.57	7.77
	LoGeR (Ours)	4.88	4.13	5.18	<u>5.03</u>	4.55	5.75	<u>8.30</u>	<u>5.40</u>
	LoGeR* (Ours)	<u>6.92</u>	<u>4.60</u>	<u>5.59</u>	3.18	<u>4.67</u>	5.48	6.44	5.27

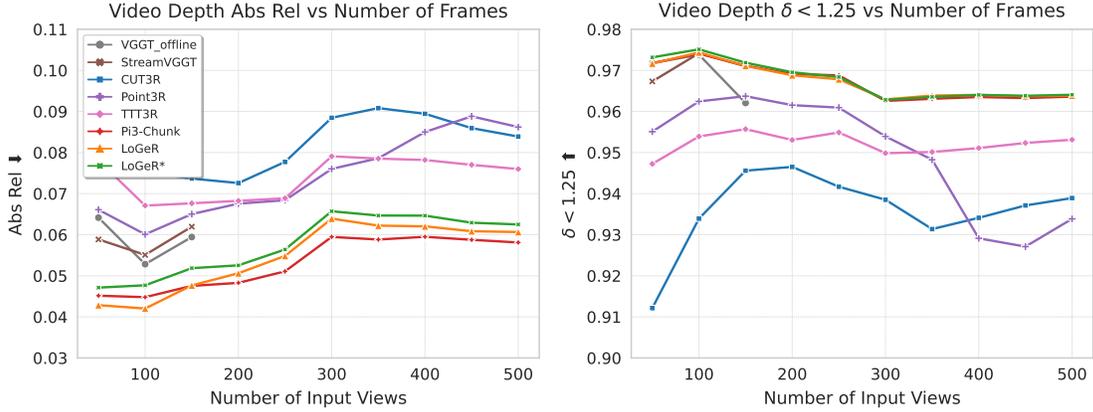


Figure 11. Quantitative results of video depth estimation on the Bonn dataset. Both our proposed baseline (Pi3-Chunk) and LoGeR significantly outperform prior works. Note that VGGT and StreamVGGT are omitted after certain frames due to out-of-memory (OOM).

Finally, similar to the strategy in LoGeR*, we compute the SE(3) alignment \mathbf{A}_m using the scale-adjusted pose of the overlapping frame k :

$$\mathbf{A}_m = \tilde{\mathbf{T}}_k^{(m-1)} (\bar{\mathbf{T}}_k^{(m)})^{-1}. \quad (15)$$

This rigid transformation is then applied to all scale-adjusted frames in the current chunk to obtain the final global camera poses:

$$\tilde{\mathbf{T}}_t^{(m)} = \mathbf{A}_m \bar{\mathbf{T}}_t^{(m)}, \quad \forall t \in \mathcal{C}_m. \quad (16)$$

B. More Experimental Results

B.1. More Evaluation on 7Scenes

Following the evaluation protocol of VGG-T³ (Elflein et al., 2026), we assess our method on the 7Scenes dataset by uniformly sampling frames across the sequences. As shown in Fig. 12, LoGeR significantly outperforms recent concurrent works across all sampled frame counts.

B.2. Depth Evaluation

We follow the evaluation protocol of TTT3R (Chen et al., 2026) to evaluate video depth on the Bonn dataset (Palazzolo et al., 2019) for sequences up to 500 frames.

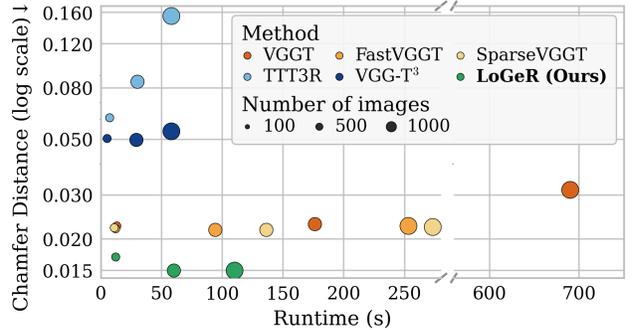


Figure 12. 3D reconstruction results on the 7-Scenes dataset, uniformly sampling 100, 500, and 1k frames. For the evaluation of 1k frames, our method achieves a **90.3%** and **72.1%** performance boost (error reduction) with comparable runtimes against concurrent works TTT3R (Chen et al., 2026) and VGG-T³ (Elflein et al., 2026), respectively. LoGeR also achieves **84.1%** faster inference speed compared to VGGT with 31.0% better performance.

Conceptually, depth estimation is a more localized task compared to global camera pose estimation or 3D reconstruction, as it relies less on strict, long-term global consistency. Nevertheless, as shown in Fig. 11, both our proposed Pi3-Chunk baseline and LoGeR significantly outperform prior methods. Specifically, LoGeR achieves a 21.05% error reduction

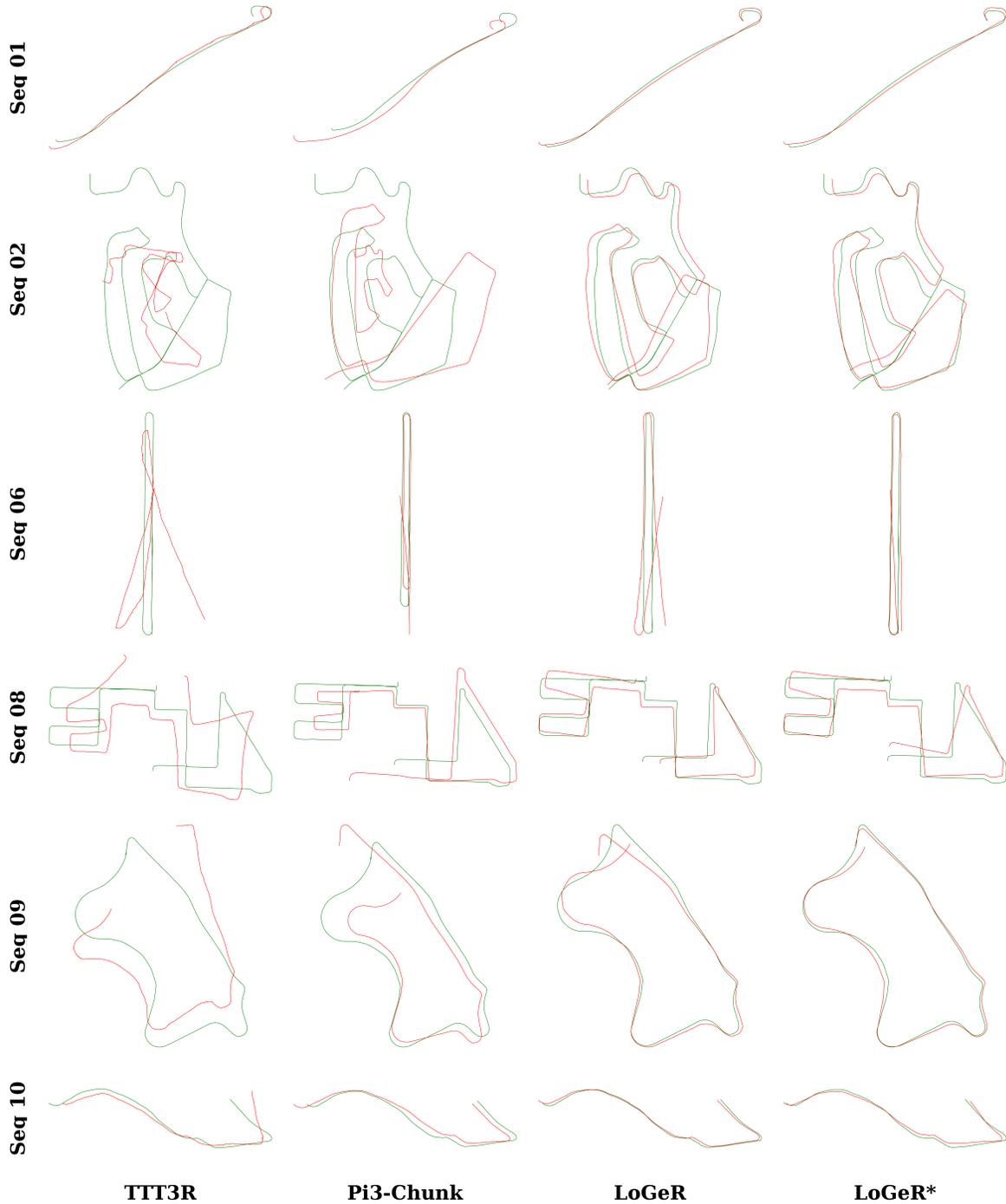


Figure 13. Qualitative results on KITTI. Ours shows more accurate long trajectory estimation.

(Abs Rel at 500 frames) compared to the previous best-performing method, TTT3R, demonstrating the efficacy of our methods.

B.3. More Detailed Results on VBR

We show in Tab. 6 the per-sequence result on VBR dataset. Our methods achieve the best on almost all sequences.

B.4. More Qualitative Results on Trajectory

We show in Fig. 13 the qualitative trajectory comparison on KITTI benchmark.

B.5. More Qualitative Results on 3D Reconstruction

Fig. 14 shows more results on large-scale indoor and outdoor scenes from in-the-wild videos and VBR.

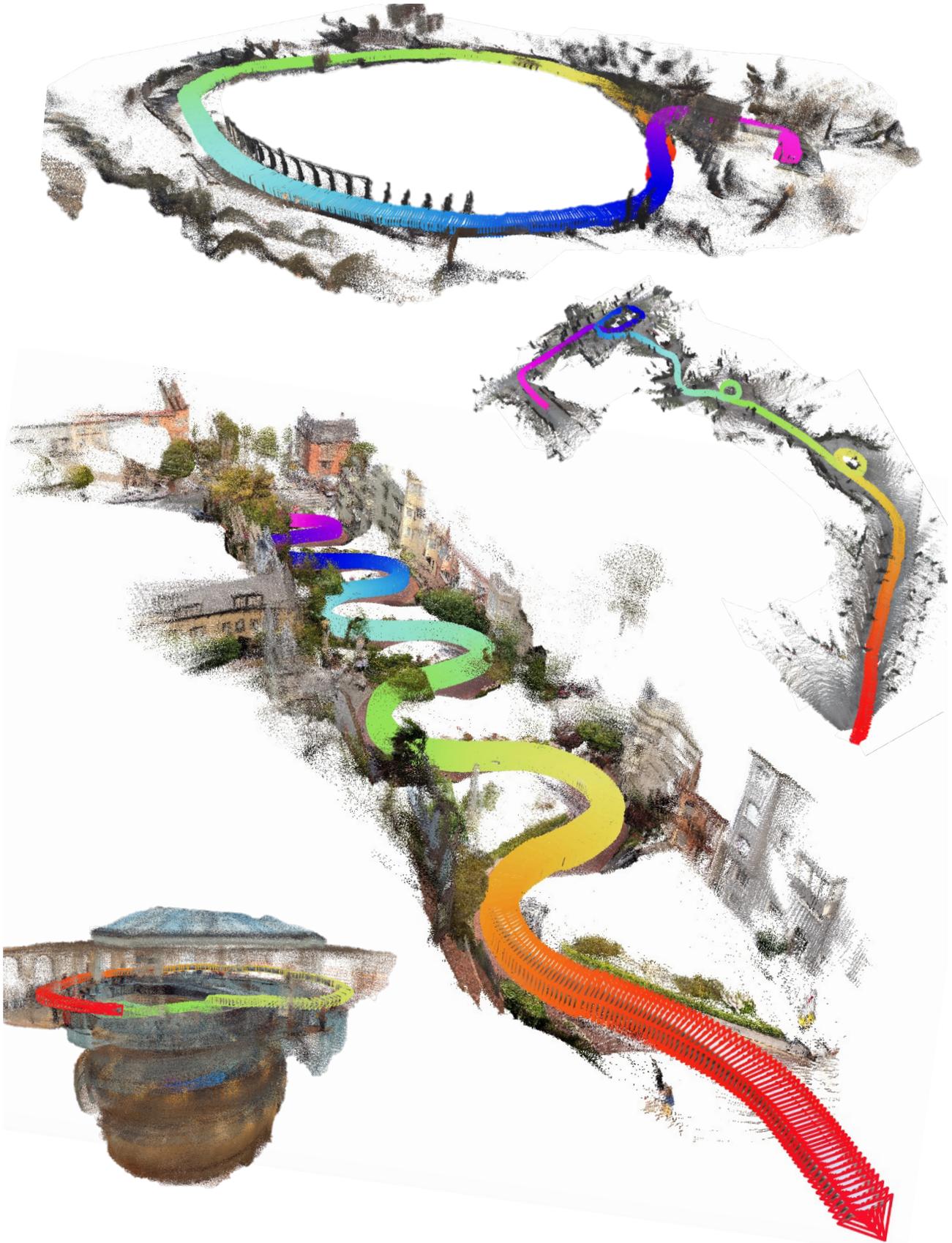


Figure 14. Qualitative results on large-scale outdoor and indoor scenes, showing minutes-long long-horizon reconstruction results.