LoGeR: Long-Context Geometric Reconstruction with Hybrid Memory

Junyi Zhang[1,2], Charles Herrmann[1,*], Junhwa Hur[1,*], Chen Sun[1], Ming-Hsuan Yang[1], Forrester Cole[1], Trevor Darrell[2], and Deqing Sun[1,†]

[1]Google DeepMind, USA [2]UC Berkeley, USA

https://LoGeR-project.github.io/

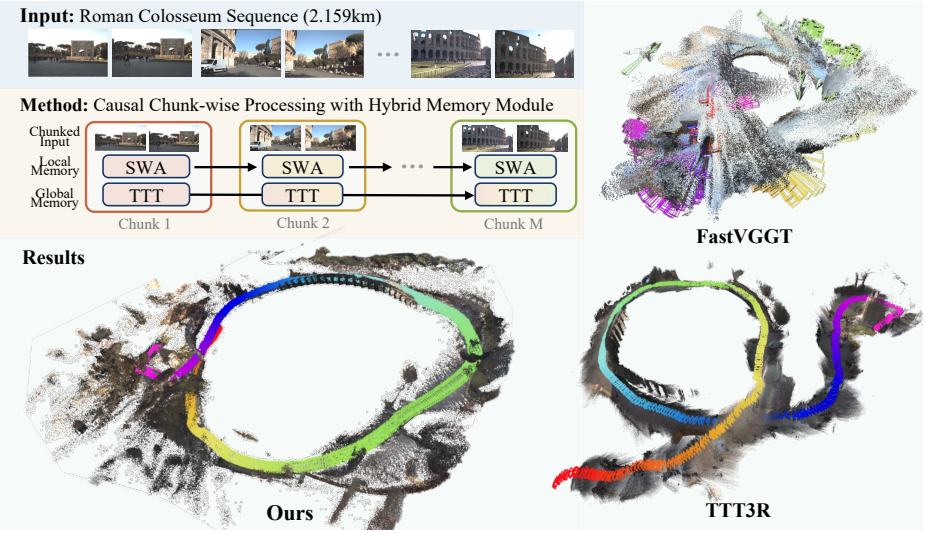

Fig. 1: Visual comparison on the large-scale VBR [4] dataset. For very long videos, we advocate for a chunk-based approach: bidirectional attention handles intra-chunk reasoning, while the proposed **hybrid memory module** ensures inter-chunk alignment. This module integrates Sliding Window Attention (SWA) for detailed local memory and Test-Time Training (TTT) for compressed global context. The proposed LoGeR achieves significant improvements over prior methods, consistently yielding superior loop closures and more precise 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** (**Lo**ng-context **Ge**ometric **R**econstruction), 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*

* Project leads. † Direction lead.

memory module. This dual-component system combines 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. Notably, this memory architecture allows LoGeR to generalize to sequences of thousands of frames during inference, despite being trained on only 128 frames. When evaluated on standard benchmarks and a repurposed VBR dataset with sequences up to 19k frames, LoGeR substantially outperforms current state-of-the-art feedforward methods. It reduces the ATE on KITTI by over 74% while achieving robust, globally consistent reconstruction across unprecedented horizons.

Keywords: Dense 3D Reconstruction · Long-Sequence Modeling

1 Introduction

Large-scale dense 3D reconstruction remains a fundamental objective in computer vision, serving as a cornerstone for applications ranging from holistic scene understanding to generative world-building. For decades, classical optimization frameworks have dominated this domain; while these methods demonstrate the capacity to reconstruct city-scale environments, they typically necessitate computationally intensive offline processing and exhibit fragility when presented with sparse or textureless inputs. Recently, the emergence of geometric foundation models—including DUSt3R [44], MonST3R [52], VGGT [41], and π^3 [46]—has initiated a paradigm shift. By distilling complex geometric priors from large-scale datasets, these models facilitate robust, feedforward inference in scenarios where classical pipelines typically fail. Nevertheless, a critical research gap persists: while classical frameworks scale to urban environments, current feedforward models remain restricted to bounded scenes.

The primary obstacle to this scaling is twofold: a fundamental "context wall" in current architectures and a problematic "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). As a result, inference-time heuristics like FastVGGT [34], which successfully mitigate memory bottlenecks, still fail to generalize to the very large-scale scenes in the VBR dataset [4], as shown in Figs. 1 and 3.

Rather than waiting for the curation of diverse, large-scale long-horizon datasets, we argue that end-to-end chunk-wise processing, where the network runs sequentially on windows or chunks of the full sequence, provides a practical strategy for mitigating the current data bottleneck. 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 [46]. However, this paradigm introduces a critical new challenge: managing memory and coherence across chunk boundaries. Achieving

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 *w.r.t.* 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**

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 [42] 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 drift.

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 [37]) that compresses global context. This hybrid design effectively decouples these tasks: the long-range parametric memory anchors the global coordinate frame to prevent drift, while the short-range non-parametric memory ensures high-precision transitions.

Furthermore, to evaluate these mechanisms at an unprecedented scale, we introduce a long-context reconstruction benchmark derived from the VBR dataset [4] by repurposing from its robotics SLAM setting to our monocular visual input setting. The repurposed dataset spans 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 addresses 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 **55.2%** relative improvement on the extremely long sequences of VBR benchmark.

2 Related Work

Learning-based visual SLAM. Recent learning-based visual SLAM methods demonstrate outperforming classical approaches [10, 26–28] by learning strong 3D priors from training datasets [20, 22, 38] or leveraging powerful pretrained visual geometry models [9, 24]. 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 [18, 41, 44, 46, 52]. 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 [7, 16, 40, 48], RNN-style persistent state [42], or efficiency-oriented VGGT variants with sparse [34, 50] or causal [55] attention. However, their lack of demonstrated results on exceptionally long sequences (*e.g.*, minute-long) leaves their scalability in sequence lengths unverified. Concurrently, TTT3R [6] introduces a confidence-based update for single-frame streaming [42]. However, this frame-wise linear approach lacks the expressivity to capture complex temporal context or leverage the powerful multi-frame reasoning of bidirectional backbones [41, 46].

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 [12, 13] and linear attention [15, 32] to reduce the quadratic cost of Transformers. A complementary line of work uses local attention (*e.g.*, sliding-window attention) [3, 51] 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) [37, 53] 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 [3] or Jamba [17]), 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, synergizing non-parametric SWA for precise adjacent alignment with parametric TTT for global consistency.

Dataset. Existing training datasets include both real (ARKitScenes, DL3DV [21], MegaDepth [19], ScanNet [8], Waymo [33]) and synthetic ones (HyperSim [30], Spring [25], TartanAir [45], TartanAirV2 [29], UnReal4K [1], Virtual KITTI 2 [5], OmniWorld-Game [54]). 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 to the expansive VBR benchmark.

3 Preliminaries

We first provide the technical background for our approach.

3D reconstruction models. Existing methods often employ bidirectional transformer architectures to fully capture local context. Given a set of N multi-view images or video frames as input $\mathbf{I} = \{I_i\}_{i=1}^{N}$, where $I_i \in \mathbb{R}^{H\times W\times 3}$, transformers predict 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 [37] 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 $\mathbf{x} = [x_1, x_2, \ldots, x_N]$ 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 $f_W(\cdot) : \mathbb{R}^d \to \mathbb{R}^d$ parameterized by the fast weights W, which are then utilized in two operations.

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

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

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

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 [14] 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.

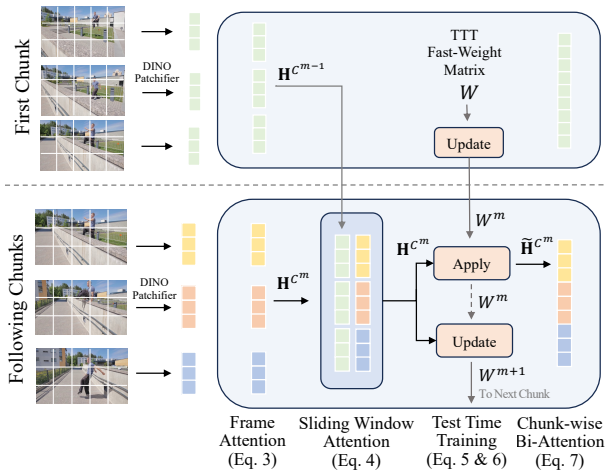

Fig. 2: Overview of a single block of our hybrid memory module. We process the input sequence in consecutive chunks of frames. While each block utilizes frame and bidirectional attention from prior work, we introduce new components to effectively propagate information across the entire sequence. Specifically, 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.

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) [53], 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 $\mathcal{C}_1^m, \ldots, \mathcal{C}_n^m$; $\mathbf{H}^{\mathcal{C}_i^m}$ be the token sequence specific to the frame $\mathcal{C}_i^m$; $\mathbf{H}^{\mathcal{C}^m}$ be the full token sequence entering a block for chunk m; and $\mathrm{LN}(\cdot)$ 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:

$$\mathbf{H}^{\mathcal{C}^m} \leftarrow \mathbf{H}^{\mathcal{C}^m} + [\mathrm{Attn}_{\mathrm{frame}}\Big(\mathrm{LN}(\mathbf{H}^{\mathcal{C}_i^m}); \theta\Big), |i \in \{1, \ldots, n\}] \tag{3}$$

where [] 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:

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

(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.

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

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

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 online 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 $\mathbf{H}^{\mathcal{C}^m}$ then serves as the input to the subsequent network block:

$$\mathbf{H}^{\mathcal{C}^m} \leftarrow \tilde{\mathbf{H}}^{\mathcal{C}^m} + \mathrm{BiAttn}_{\mathrm{chunk}}\Big(\mathrm{LN}(\tilde{\mathbf{H}}^{\mathcal{C}^m});\, \theta\Big). \tag{7}$$

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 [46] and train LoGeR with (i) a *scale-invariant local pointmap loss* and (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 [43], 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.

$$\mathcal{L}_{\mathrm{local}} = \frac{1}{N|\Omega|}\sum_{i=1}^{N}\sum_{p\in\Omega}\frac{1}{z_{i,p}}\left\| s^*\hat{\mathbf{x}}_{i,p} - \mathbf{x}_{i,p}\right\|_1, \tag{8}$$

$$\mathcal{L}_{\mathrm{pose}} = \sum_{(i,j)\in\mathcal{P}}\Big(\lambda_{\mathrm{r}}\mathcal{L}_{\mathrm{rot}}(\hat{\mathbf{R}}_{ij}, \mathbf{R}_{ij}) + \lambda_{\mathrm{t}}\left\| s^*\hat{\mathbf{t}}_{ij} - \mathbf{t}_{ij}\right\|_{\mathrm{Huber}}\Big), \tag{9}$$

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

$$\mathcal{L} = \mathcal{L}_{\mathrm{local}} + \mathcal{L}_{\mathrm{pose}} + \lambda_{\mathrm{global}}\mathcal{L}_{\mathrm{global}}. \tag{11}$$

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

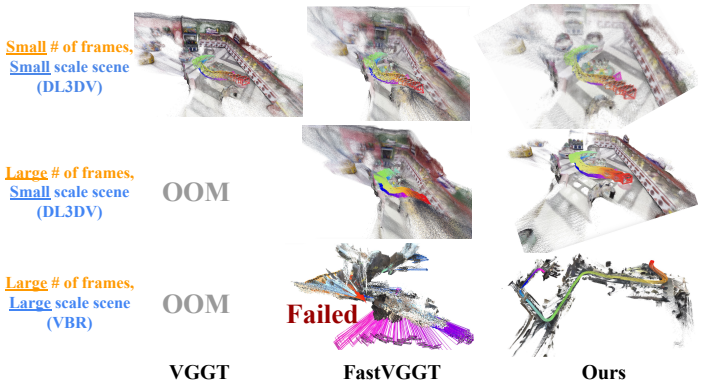

Fig. 3: Comparison of different methods across varying sequence lengths and scene scales. Although FastVGGT is able to process a larger number of frames during inference, it fails completely on large-scale scenes, highlighting the inherent "data wall" of models trained strictly on short-context bubbles.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

for normalization. $\hat{\mathbf{T}}_i = [\hat{\mathbf{R}}_i \mid \hat{\mathbf{t}}_i] \in \mathrm{SE}(3)$ 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 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 $\mathrm{SE}(3)$ 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: $\mathbf{A}_m = \tilde{\mathbf{T}}_k^{(m-1)} \big(\hat{\mathbf{T}}_k^{(m)}\big)^{-1}$. This transformation is applied to all frames in $\mathcal{C}^m$:

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

We use $\tilde{\mathbf{T}}_t^{(m)}$ as the final 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 [41], even when equipped with inference-time architectural efficiency improvement (FastVGGT [34]), fail to generalize to large-scale scenes when trained solely on short-context or small-scene-scale data. To overcome this

Table 2: Comparison of Absolute Trajectory Error (ATE↓, m) on KITTI. The top and bottom blocks denote *optimization-based* and *feedforward* methods, respectively. Red and orange cells 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.**
	num. of frames, scale	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	-
	Contains Loop?	✓	×	✓	×	×	✓	✓	✓	×	✓	×	-
Opt.-based	DROID-SLAM [38]	92.10	344.60	107.61	2.38	1.00	118.50	62.47	21.78	161.60	72.32	118.70	100.28
Opt.-based	DPV-SLAM [22]	112.80	11.50	123.53	2.50	0.81	57.80	54.86	18.77	110.49	76.66	13.65	53.03
Opt.-based	DPV-SLAM++ [22]	8.30	11.86	39.64	2.50	0.78	5.74	11.60	1.52	110.90	76.70	13.70	25.75
Opt.-based	VGGT-Long [9]	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 [34]	OOM	639.39	OOM	21.53	9.51	OOM	40.56	51.35	OOM	201.54	196.22	–
Feedforward	InfiniteVGGT [50]	186.46	623.62	289.16	166.74	68.00	143.84	117.57	85.33	221.56	215.41	156.92	206.78
Feedforward	CUT3R [42]	190.38	90.59	264.39	20.40	7.31	92.25	67.54	22.48	145.08	67.42	40.00	91.62
Feedforward	TTT3R [6]	119.94	99.59	238.07	16.83	3.98	36.38	47.20	11.62	107.33	86.96	33.58	72.86
Feedforward	Pi3-Chunk (Proposed Baseline)	26.65	196.04	157.92	5.13	1.09	12.79	27.66	5.94	61.26	56.31	21.96	52.07
Feedforward	**LoGeR (Ours)**	62.34	41.64	39.64	4.89	1.82	41.27	13.99	16.24	26.46	22.71	8.84	25.44
Feedforward	**LoGeR* (Ours)**	30.47	47.91	36.32	5.38	1.95	26.34	6.60	5.55	24.41	10.12	10.11	18.65

"data wall", we construct a training mixture heavily weighted towards large-scale-scene datasets, *e.g.*, TartanAirV2 [29], 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 Sec. 5.3.

5 Experiments

Implementation details. We train our model on a mixture of real and synthetic datasets, including ARKitScenes [2], DL3DV [21], HyperSim [30], MegaDepth [19], ScanNet [8], ScanNet++ [49], Spring [25], TartanAir [45], TartanAirV2 [29], UnReal4K [47], Virtual KITTI 2 [5], Waymo [33], and a subset of OmniWorld-Game. We optimize the model with AdamW [23] 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 weights of the patchifier, frame attention, and chunk-wise bidirectional attention modules from π^3 [46]. All the evaluation is done on a NVIDIA A100 40GB. See more details in Appendix ??????.

5.1 Evaluation on Long Sequences

Benchmarks. As 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. We first use **KITTI benchmark** [11], 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** [4]. 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 [39], following standard protocols [22, 38].

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 [38] and DPV-SLAM [22], as well as recent foundation-model-based approaches like VGGT-SLAM and VGGT-Long [41]. For *feedforward* methods, we evaluate recurrent architectures including CUT3R [42] and TTT3R [6], alongside bidirectional models such as FastVGGT [34] and autoregressive approaches like InfiniteVGGT [50]. 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 [31]. Note, we also apply the feedforward pose alignment when doing a reset. See Appendix **??** 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 **??**.

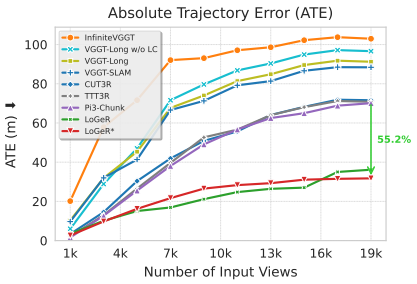

Fig. 4: Quantitative results on our proposed VBR [4] evaluation showing results on very long sequences spanning from 1,000 to 19,000 frames. Our methods achieve 55.2% more accurate results than prior methods.

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 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

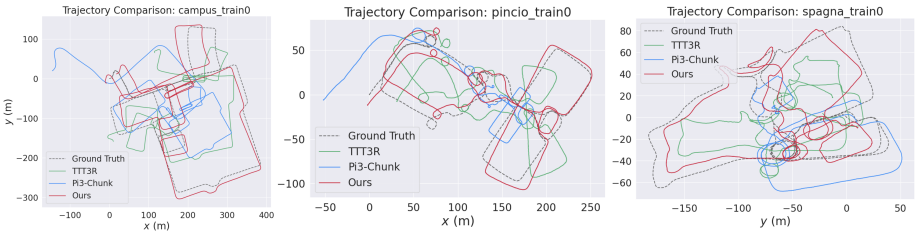

Fig. 5: Qualitative camera trajectories on the VBR dataset. LoGeR accurately preserves global scale and trajectory over very long sequences, closely matching the ground truth where prior methods suffer from severe drift.

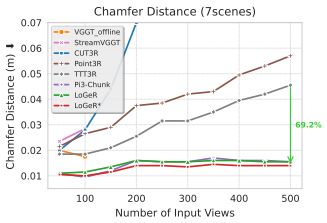

Fig. 6: 3D reconstruction result on 7scenes. Both our proposed baseline and LoGeR significantly outperform prior work by **69.2%**.

Fig. 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.

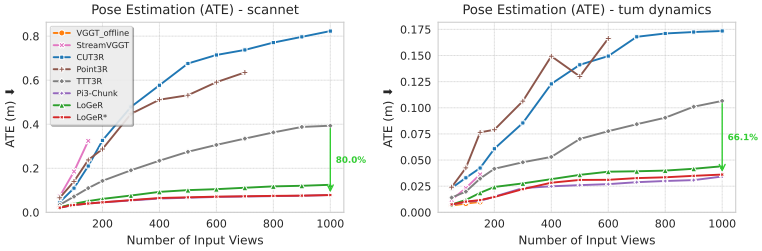

Fig. 8: Quantitative results on camera pose evaluation, demonstrating that both our proposed baseline and LoGeR substantially outperform prior work, with **80.0%** and **66.1%** relative gains on ScanNet and TUM datasets, respectively.

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 results: LoGeR maintains global scale consistency up to 20k frames, whereas the baseline exhibits severe scale drift in such extremely long sequences. See Appendix **??????** for more 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 [35] with

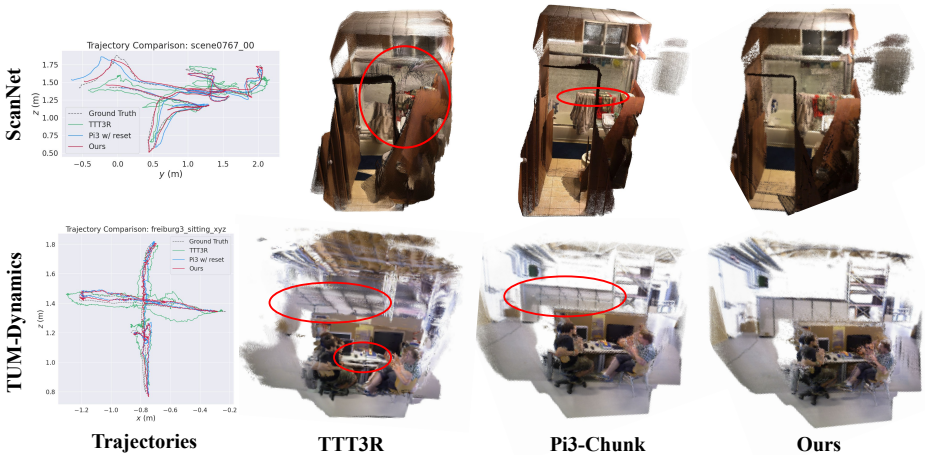

Fig. 9: Qualitative results on ScanNet and TUM-Dynamics. While the proposed Pi3-Chunk baseline yields slightly better pose metrics on small-scale TUM sequences (unlike its severe drift on longer trajectories), LoGeR produces visually superior reconstructions. As highlighted, LoGeR accurately recovers structural details, avoiding the severe distortions and geometric artifacts present in baselines.

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

sequence lengths ranging from 50 to 500 frames. Subsequently, we evaluate camera pose estimation on ScanNetV2 [8] and TUM-Dynamics [36] 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 **????** for more results.

5.3 Ablation Study

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

Table 3: Ablation results on the ScanNet (subset) and TUM datasets. We report the results of ATE (↓) on both 500 and 1000 frames of sequences.

	ScanNet		TUM	
Method	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

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 presented LoGeR, a hybrid architecture that integrates Sliding Window Attention (SWA) with a Test-Time Training (TTT) layer to overcome the challenges of long-sequence estimation in feedforward 3D geometry models. By leveraging the synergy between these complementary mechanisms, LoGeR efficiently processes thousands of frames without per-sequence optimization. Our approach consistently outperforms existing state-of-the-art methods—including those based on explicit memory, recurrent states, or causal attention—on the challenging long-sequence VBR benchmark. We hope that LoGeR could serve as a catalyst for new developments in long-context spatio-temporal reasoning, video understanding, and robotic perception.

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 the help with the training infrastructure. We thank Xingyu Chen for providing details on the evaluation. We thank Zihan Zhu for his feedback on the evaluation of the VBR benchmark. We thank Angjoo Kanazawa, Tyler Bonnen, and Jiahui Lei for helpful feedback on the manuscript.

References

1. Aleotti, F., Tosi, F., Ramirez, P.Z., Poggi, M., Salti, S., Mattoccia, S., Di Stefano, L.: Neural disparity refinement for arbitrary resolution stereo. In: 3DV. pp. 207–217. IEEE (2021) 4
2. 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) 10
3. Beltagy, I., Peters, M.E., Cohan, A.: Longformer: The long-document transformer. arXiv preprint arXiv:2004.05150 (2020) 4
4. Brizi, L., Giacomini, E., Di Giammarino, L., Ferrari, S., Salem, O., De Rebotti, L., Grisetti, G.: VBR: A vision benchmark in Rome. In: ICRA. pp. 15868–15874. IEEE (2024) 1, 2, 3, 11
5. Cabon, Y., Murray, N., Humenberger, M.: Virtual KITTI 2. arXiv preprint arXiv:2001.10773 (2020) 4, 10
6. Chen, X., Chen, Y., Xiu, Y., Geiger, A., Chen, A.: TTT3R: 3D reconstruction as test-time training. In: ICLR (2026) 4, 10, 11
7. Chen, Z., Qin, M., Yuan, T., Liu, Z., Zhao, H.: Long3R: Long sequence streaming 3D reconstruction. In: ICCV. pp. 5273–5284 (2025) 4
8. Dai, A., Chang, A.X., Savva, M., Halber, M., Funkhouser, T., Nießner, M.: ScanNet: Richly-annotated 3D reconstructions of indoor scenes. In: CVPR. pp. 5828–5839 (2017) 4, 10, 13
9. Deng, K., Ti, Z., Xu, J., Yang, J., 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) 3, 10
10. Engel, J., Schöps, T., Cremers, D.: LSD-SLAM: Large-scale direct monocular SLAM. In: ECCV. pp. 834–849 (2014) 3
11. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the KITTI vision benchmark suite. In: CVPR. pp. 3354–3361. IEEE (2012) 11
12. Gu, A., Dao, T.: Mamba: Linear-time sequence modeling with selective state spaces. In: COLM (2024) 4
13. Gu, A., Goel, K., Re, C.: Efficiently modeling long sequences with structured state spaces. In: ICLR (2022) 4
14. Jordan, K., Jin, Y., Boza, V., Jiacheng, Y., Cesista, F., Newhouse, L., Bernstein, J.: Muon: An optimizer for hidden layers in neural networks (2024) 5
15. Katharopoulos, A., Vyas, A., Pappas, N., Fleuret, F.: Transformers are RNNs: Fast autoregressive transformers with linear attention. In: ICML. pp. 5156–5165. PMLR (2020) 4

16. Lan, Y., Luo, Y., Hong, F., Zhou, S., Chen, H., Lyu, Z., Yang, S., Dai, B., Loy, C.C., Pan, X.: Stream3R: Scalable sequential 3D reconstruction with causal transformer. arXiv preprint arXiv:2508.10893 (2025) 4
17. 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) 4
18. Leroy, V., Cabon, Y., Revaud, J.: Grounding image matching in 3D with MASt3R. In: ECCV. pp. 71–91 (2024) 4
19. Li, Z., Snavely, N.: MegaDepth: Learning single-view depth prediction from internet photos. In: CVPR. pp. 2041–2050 (2018) 4, 10
20. Li, Z., Tucker, R., Cole, F., Wang, Q., Jin, L., Ye, V., Kanazawa, A., Holynski, A., Snavely, N.: MegaSaM: Accurate, fast and robust structure and motion from casual dynamic videos. In: CVPR. pp. 10486–10496 (2025) 3
21. 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) 4, 10
22. Lipson, L., Teed, Z., Deng, J.: Deep patch visual SLAM. In: ECCV. pp. 424–440. Springer (2024) 3, 10, 11
23. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: ICLR (2019) 10
24. Maggio, D., Lim, H., Carlone, L.: VGGT-SLAM: Dense RGB SLAM optimized on the SL (4) manifold. arXiv preprint arXiv:2505.12549 (2025) 3
25. Mehl, L., Schmalfuss, J., Jahedi, A., Nalivayko, Y., Bruhn, A.: Spring: A high-resolution high-detail dataset and benchmark for scene flow, optical flow and stereo. In: CVPR. pp. 4981–4991 (2023) 4, 10
26. Mur-Artal, R., Montiel, J.M.M., Tardos, J.D.: ORB-SLAM: a versatile and accurate monocular SLAM system. IEEE Transactions on Robotics **31**(5), 1147–1163 (2015) 3
27. Mur-Artal, R., 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) 3
28. Newcombe, R.A., Lovegrove, S.J., Davison, A.J.: DTAM: Dense tracking and mapping in real-time. In: ICCV. pp. 2320–2327 (2011) 3
29. Patel, M., Yang, F., Qiu, Y., Cadena, C., Scherer, S., Hutter, M., Wang, W.: TartanGround: A large-scale dataset for ground robot perception and navigation. In: IROS (2025) 4, 10
30. Roberts, M., Ramapuram, J., Ranjan, A., Kumar, A., Bautista, M.A., Paczan, N., Webb, R., Susskind, J.M.: Hypersim: A photorealistic synthetic dataset for holistic indoor scene understanding. In: ICCV. pp. 10912–10922 (2021) 4, 10
31. Ruiz, R.B., Gu, A.: Understanding and improving length generalization in recurrent models. In: ICML (2025) 11
32. Schlag, I., Irie, K., Schmidhuber, J.: Linear transformers are secretly fast weight programmers. In: ICML. pp. 9355–9366. PMLR (2021) 4
33. Schwall, M., Daniel, T., Victor, T., Favaro, F., Hohnhold, H.: Waymo public road safety performance data. arXiv preprint arXiv:2011.00038 (2020) 4, 10
34. Shen, Y., Zhang, Z., Qu, Y., Cao, L.: FastVGGT: Training-free acceleration of visual geometry transformer. In: ICLR (2026) 2, 4, 9, 10, 11
35. Shotton, J., Glocker, B., Zach, C., Izadi, S., Criminisi, A., Fitzgibbon, A.: Scene coordinate regression forests for camera relocalization in RGB-D images. In: CVPR. pp. 2930–2937 (2013) 12

36. Sturm, J., Engelhard, N., Endres, F., Burgard, W., Cremers, D.: A benchmark for the evaluation of RGB-D SLAM systems. In: IROS. pp. 573–580 (2012) 13
37. 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) 3, 4, 5
38. Teed, Z., Deng, J.: DROID-SLAM: Deep visual SLAM for monocular, stereo, and RGB-D cameras. NeurIPS pp. 16558–16569 (2021) 3, 10, 11
39. Umeyama, S.: Least-squares estimation of transformation parameters between two point patterns. IEEE TPAMI **13**(4), 376–380 (1991) 11
40. Wang, H., Agapito, L.: 3D reconstruction with spatial memory. In: 3DV (2025) 4
41. Wang, J., Chen, M., Karaev, N., Vedaldi, A., Rupprecht, C., Novotny, D.: VGGT: Visual geometry grounded transformer. In: CVPR. pp. 5294–5306 (2025) 2, 4, 9, 11
42. Wang, Q., Zhang, Y., Holynski, A., Efros, A.A., Kanazawa, A.: Continuous 3D perception model with persistent state. In: CVPR. pp. 10510–10522 (2025) 3, 4, 10, 11
43. Wang, R., Xu, S., Dai, C., Xiang, J., Deng, Y., Tong, X., Yang, J.: MoGe: Unlocking accurate monocular geometry estimation for open-domain images with optimal training supervision. In: CVPR. pp. 5261–5271 (2025) 8
44. Wang, S., Leroy, V., Cabon, Y., Chidlovskii, B., Revaud, J.: DUSt3R: Geometric 3D vision made easy. In: CVPR. pp. 20697–20709 (2024) 2, 4
45. Wang, W., Zhu, D., Wang, X., Hu, Y., Qiu, Y., Wang, C., Hu, Y., Kapoor, A., Scherer, S.: TartanAir: A dataset to push the limits of visual SLAM. In: IROS. pp. 4909–4916 (2020) 4, 10
46. Wang, Y., Zhou, J., Zhu, H., Chang, W., Zhou, Y., Li, Z., Chen, J., Pang, J., Shen, C., He, T.: π^3: Permutation-equivariant visual geometry learning. In: ICLR (2026) 2, 4, 8, 10
47. Wang, Y., Shi, M., Li, J., Huang, Z., Cao, Z., Zhang, J., Xian, K., Lin, G.: Neural video depth stabilizer. In: ICCV. pp. 9466–9476 (2023) 10
48. Wu, Y., Zheng, W., Zhou, J., Lu, J.: Point3R: Streaming 3D reconstruction with explicit spatial pointer memory. In: NeurIPS (2025) 4
49. Yeshwanth, C., Liu, Y.C., Nießner, M., Dai, A.: Scannet++: A high-fidelity dataset of 3D indoor scenes. In: ICCV. pp. 12–22 (2023) 10
50. Yuan, S., Yang, Y., Yang, X., Zhang, X., Zhao, Z., Zhang, L., Zhang, Z.: InfiniteVGGT: Visual geometry grounded transformer for endless streams. arXiv preprint arXiv:2601.02281 (2026) 4, 10, 11
51. 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) 4
52. Zhang, J., Herrmann, C., Hur, J., Jampani, V., Darrell, T., Cole, F., Sun, D., Yang, M.H.: MonST3R: A simple approach for estimating geometry in the presence of motion. In: ICLR (2025) 2, 4
53. Zhang, T., Bi, S., Hong, Y., Zhang, K., Luan, F., Yang, S., Sunkavalli, K., Freeman, W.T., Tan, H.: Test-time training done right. arXiv preprint arXiv:2505.23884 (2025) 4, 7
54. 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) 4
55. Zhuo, D., Zheng, W., Guo, J., Wu, Y., Zhou, J., Lu, J.: Streaming 4D visual geometry transformer. In: ICLR (2026) 4