HorizonStream: Long-Horizon Attention for Streaming 3D Reconstruction

Chong Cheng[1,2] Peilin Tao[2,3] Nanjie Yao[1] Guanzhi Ding[1] Xianda Chen[4] Yuansen Du[2]
Xiaoyang Guo[2] Wei Yin[2] Weiqiang Ren[2] Qian Zhang[2] Zhengqing Chen[2,‡] Hao Wang[1,†]
[1]HKUST(GZ) [2]Horizon Robotics [3]CASIA [4]CSU
† Corresponding author ‡ Project lead

Abstract

Online 3D reconstruction requires estimating camera pose and scene geometry under strict causal and bounded-memory constraints. Existing methods often suffer from drift, jitter, or collapse on long sequences. We trace these failures to a fundamental mismatch. Streaming geometry is inherently temporally heterogeneous, with evidence ranging from short-lived correspondences to persistent global scale. However, current architectures impose uniform and pathological influence patterns. For example, sliding windows enforce hard cutoffs, while ungated recurrence and causal attention cause cache saturation and spike-like attention sinks. To resolve this, we formalize geometric propagation as an *evidence influence kernel* and propose HorizonStream, a long-horizon Transformer that explicitly factorizes this kernel. For the long-range temporal factor, Geometric Linear Attention learns channel-wise decay rates to enable bounded, multi-timescale propagation of geometric evidence. For the short-range spatial factor, Gated Window Attention (GWA) with Spatiotemporal RoPE performs reliable 3D matching while suppressing attention sinks. Finally, Metric Readout Tokens recover stable scale and rigid pose directly from the persistent geometric state. Extensive experiments show that HorizonStream, trained on only 48-frame clips, generalizes stably to sequences exceeding $10{,}000$ frames with constant memory and linear time, achieving state-of-the-art streaming 3D reconstruction performance. Project Page: `https://3dagentworld.github.io/horizonstream/`

1 Introduction

Online 3D reconstruction from streaming video is a core capability for robotics, autonomous driving, and embodied intelligence, requiring causal, bounded-memory estimation of camera pose and scene geometry. Optimization-based methods [32, 42, 4, 43, 8, 36] maintain explicit geometric states, but rely on iterative optimization and have limited throughput. Recent offline feed-forward methods [47, 21, 45, 34, 12, 61, 17, 7, 13, 14] achieve high accuracy, but use full attention and access future frames, violating online causality.

Strictly causal streaming 3D reconstruction still degrades on long sequences [11]. Methods often suffer from collapse, pose jitter, and scale instability. This occurs because existing architectures organize history purely by recency. However, recency is a poor proxy for geometric relevance in 3D, as streaming geometry is inherently temporally heterogeneous. Recent evidence may already be invalid, while older evidence can remain reliable. Therefore, we view the reconstruction process as aggregating diverse types of geometric ***evidence***. This evidence has vastly different lifetimes. For example, local 2D-3D correspondences are short-lived, which quickly become invalid due to motion. In contrast, global scale and scene structures are persistent, which must remain reliable over long horizons. Yet, existing architectures impose a uniform propagation rule on all evidence. The

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

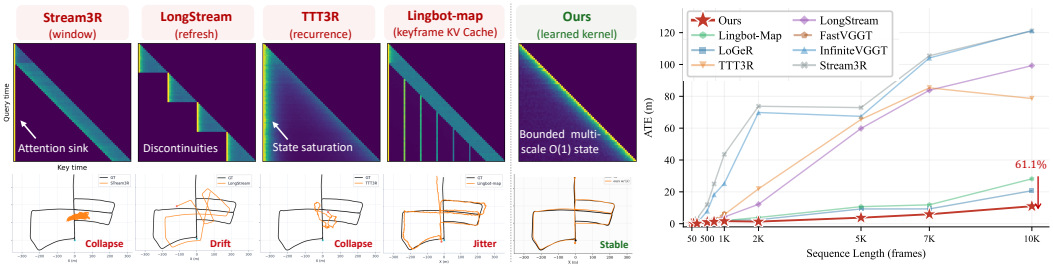


Figure 1: Geometric evidence influence patterns on KITTI and long-sequence scaling on VBR. Prior streaming methods impose hard cutoffs, refresh discontinuities, heavy-tailed states, or spiky KV caches, leading to pose degradation, jitter, or collapse. HorizonStream learns a bounded multi-scale kernel with an $O(1)$ recurrent geometric state and maintains stable ATE up to 10K frames.

key question is: *how can we apply the correct temporal influence range for each type of geometric evidence?*

To answer this, we further formalize the temporal propagation of geometric information through an ***evidence influence kernel***. We define this kernel as a spatio-temporal weight function, which determines how much past geometric evidence should influence the current reconstruction state. Under this formulation, we find that existing methods inadvertently induce pathological kernels, as shown in Fig. 1. Sliding windows [20, 65] impose a hard-cutoff box kernel, which may prematurely discard useful past evidence. Refresh mechanisms [11, 9] create blockwise discontinuous kernels. Causal softmax attention [5] degenerates into spike-like attention sinks, which focus on irrelevant early tokens. Ungated recurrence [6, 46] forms a heavy-tailed kernel with unbounded error accumulation. As sequences grow longer, these pathological kernels are repeatedly amplified. This causes cache saturation, early-token dominance, and severe geometric drift.

Consequently, current geometric transformer memory designs occupy two extremes of a retention spectrum. Sliding windows force immediate forgetting. Full-attention methods retain everything permanently. Both extremes lack a bounded, flexible temporal form. Instead, a proper approach should learn continuous retention rates tailored to each geometric channel.

To this end, we propose **HorizonStream**, a long-horizon Transformer that explicitly instantiates this kernel factorization. For the long-range temporal factor, **Geometric Linear Attention** maintains a bounded $O(1)$ recurrent state derived from a discounted geometric objective. By learning channel-wise exponential decay rates, it enables stable multi-timescale evidence propagation across windows. For the short-range spatial factor, **Gated Window Attention** performs 3D content matching within the local window. It uses head-wise reliability gates to filter noisy correspondences and suppress attention sinks, while spatiotemporal RoPE provides relative 3D space-time position bias. Finally, to satisfy the metric invariance constraint, Metric Readout Tokens (MRT) and relative pose fusion recover stable scale and rigid pose directly from the persistent recurrent state.

Since the proposed kernel is local and bounded, it defines a sequence-length-independent propagation rule that can be repeatedly applied to arbitrary-length streams. Experiments on multiple datasets show that HorizonStream, trained on only 48-frame clips, generalizes stably to tens of thousands of frames without pose degradation and outperforms all streaming 3D reconstruction methods.

Our contributions are:

- We formalize streaming 3D reconstruction via a geometric evidence influence kernel. This view unifies common long-sequence failures as pathological kernel shapes, i.e., hard cutoffs, discontinuities, attention sinks, and cache saturation.
- We propose HorizonStream, a constrained kernel-decomposition architecture. Geometric Linear Attention provides bounded multi-timescale propagation across windows; Gated Window Attention with Spatiotemporal RoPE enables content-aware 3D matching within windows; MRT with relative pose fusion preserves metric scale and rigid pose.
- Experiments on multiple datasets show that HorizonStream, trained only with 48-frame batches, generalizes to sequences over 10,000 frames with constant memory and linear time, achieving state-of-the-art streaming 3D reconstruction performance.



Figure 2: Visualization of long-range streaming 3D reconstruction across diverse scenes. Our method maintains stable trajectories and coherent geometry over sequences ranging from hundreds to thousands of frames in outdoor, indoor, and game environments.

2 Related Work

Optimization-based visual SLAM and odometry. Classical visual SLAM systems such as ORB-SLAM3 [4] maintain explicit geometric maps using keyframe management, local optimization, and loop closure. DROID-SLAM and DPVO [42, 43] combine learned correspondence estimation with iterative bundle adjustment. MASt3R-SLAM and VGGT-SLAM [27, 26] integrate feed-forward reconstruction priors into tracking, map fusion or submap alignment, and global optimization.

Offline feed-forward 3D reconstruction. DUSt3R [47, 57] and MASt3R [21, 10] predict dense geometry from image pairs. This paradigm extends to sequences via spatial memory in Spann3R and MonST3R [44, 60], and to arbitrary image collections with a geometry-aware Transformer in VGGT [45]. FastVGGT [34] reduces inference memory by reusing attention maps. VGGT-Long [12] and LoGeR [61] scale to longer inputs through chunk-wise processing or accumulated weights. However, they usually rely on full attention within chunks and chunk stitching lacks cross-chunk dependency, causing temporal discontinuities.

Online feed-forward 3D reconstruction. Recent methods adapt feed-forward reconstruction to causal streams [58]. STream3R and StreamVGGT [20, 65] use causal masks and sliding-window attention, retaining only local-window context. CUT3R and TTT3R [46, 6] add persistent recurrent states, Point3R [51] maintains spatial pointer memory, InfiniteVGGT [59] prunes the KV cache, and Lingbot-map [5] extends context with keyframe memory. These designs enable cross-window information transfer but rely on fixed or write-only temporal mechanisms and still suffer from jitter, pose degradation, and disordered geometry on long sequences. LongStream [11] attributes long-sequence degradation to attention sink and state saturation [53, 16], but its periodic cache refresh discards accumulated context at each boundary, weakening long-range revisit.

Therefore, we argue that a better online 3D reconstruction pipeline requires a bounded and multi-timescale control over geometric evidence influence. HorizonStream learns channel-wise propagation scales to preserve useful long-range geometry and down-weight stale evidence without cache reset.

3 Method

Overview. Fig. 3 shows the HorizonStream framework. The model processes the most recent W frames causally and maintains an $O(1)$ geometric state for cross-window structure and scale. Gated Window Attention with Spatiotemporal RoPE handles within-window matching, Geometric Linear Attention performs cross-window propagation, and Metric Readout Tokens recover scale.

3.1 Problem Formulation

Given an RGB video, streaming 3D reconstruction predicts pose $\hat{\mathbf{T}}_t \in SE(3)$ and dense depth $\hat{D}_t$ online from past observations and a bounded state. We describe how past evidence affects the current

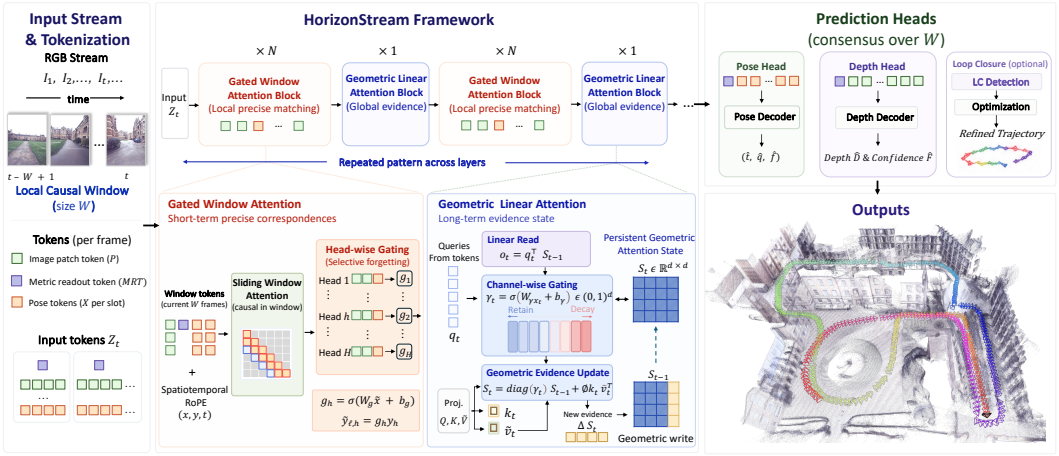


Figure 3: Overview of **HorizonStream**. Given an RGB stream, the model causally processes the most recent W frames. Gated Window Attention handles local matching, Geometric Linear Attention propagates long-range geometry with an $O(1)$ recurrent geometric state, and Metric Readout Tokens recover stable scale and pose. An optional loop-closure module refines the trajectory.

reconstruction with a geometric evidence influence kernel $K(t, i)$, which maps evidence at time i to its contribution at time t.

A valid geometric evidence influence kernel must solve three core problems: 1) select reliable local correspondences based on spatial content, 2) ensure bounded multi-timescale propagation to prevent state accumulation while respecting diverse evidence lifetimes, and 3) preserve scale and rigid pose.

To systematically address these requirements, we decouple the influence mechanism into a spatio-temporal kernel factorization augmented by a metric readout. We factorize the kernel as:

$$K(t, i) = K_{\text{spatial}}(t, i) \cdot K_{\text{time}}(t, i). \tag{1}$$

This factorization explicitly maps the three problems to dedicated computational components. First, K_{spatial} addresses spatial content-awareness (Problem 1). It uses image content and 3D proximity to select reliable short-range evidence. Second, K_{time} addresses bounded multi-timescale propagation (Problem 2). It uses channel-wise exponential decay to keep long-range influence bounded while allowing different geometric channels to propagate over distinct temporal horizons. Finally, Metric Readout Tokens read metric information from the recurrent state to recover stable scale and rigid pose (Problem 3).

Together, these components form a complete, strictly causal streaming architecture. We now detail how this theoretical framework is instantiated into our network architecture. Section 3.2 introduces Geometric Linear Attention to model the temporal factor K_{time}. Section 3.3 introduces Gated Window Attention to model the spatial factor K_{spatial}. Analysis of why open-form operators fail these constraints is provided in Appendix A and B.

3.2 Geometric Linear Attention

The long-range temporal factor functions as an online geometric estimator over key-value encoded geometric evidence, including correspondence, motion, structure, and scale cues. It summarizes this evidence in a bounded cross-window state, revises stale information, and preserves long-lived geometry. We formulate this through a discounted geometric state-estimation objective:

$$\mathcal{J}_t(\mathbf{S}) = \sum_{i=1}^{t} \Big(\prod_{j=i+1}^{t} \gamma_j \Big) \|\mathbf{S}^\top \mathbf{k}_i - \mathbf{v}_i\|_2^2, \qquad K_{\text{time}}(t, i) = \prod_{j=i+1}^{t} \gamma_j. \tag{2}$$

Here, $\mathbf{S} \in \mathbb{R}^{d \times d}$ is the recurrent geometric state. The vectors $\mathbf{k}_i$ and $\mathbf{v}_i$ are the key and value encoding the geometric evidence at time i. The variable γ acts as a learned gating factor for information retention. Specifically, γ_t denotes the retention rate at time index t, and γ_j represents the intermediate retention rate at a specific step j within the cumulative product. With $\gamma_t \equiv 1$, evidence never

decays. This causes heavy-tailed accumulation and state contamination. With $\bar{\gamma} = \sup_t |\gamma_t| < 1$, the influence of stale evidence is strictly bounded:

$$\left\| \mathbf{q}_t^\top \Big(\prod_{j=1}^{t} \gamma_j \Big) \mathbf{S}_0 \right\| \leq \|\mathbf{q}_t\| \cdot \|\mathbf{S}_0\|_F \cdot \bar{\gamma}^{\,t} \to 0. \tag{3}$$

In this bound, $\mathbf{q}_t$ is the query vector at time t, and $\mathbf{S}_0$ is the initial state. The term $\|\cdot\|_F$ denotes the Frobenius norm. Thus, discounting closes the open-form temporal influence that causes unbounded accumulation.

Online state update. The objective admits the recursive form

$$\mathcal{J}_t(\mathbf{S}) = \gamma_t \mathcal{J}_{t-1}(\mathbf{S}) + \|\mathbf{S}^\top \mathbf{k}_t - \mathbf{v}_t\|_2^2. \tag{4}$$

This principle yields a fixed-state attention update:

$$\mathbf{S}_t = \gamma_t \mathbf{S}_{t-1} + \phi(\mathbf{k}_t)\tilde{\mathbf{v}}_t^\top, \qquad \mathbf{o}_t = \mathbf{q}_t^\top \mathbf{S}_t. \tag{5}$$

Here $\mathbf{S}_t \in \mathbb{R}^{d \times d}$ summarizes cross-window reconstruction evidence, $\phi(\mathbf{k}_t)$ maps keys into the linear attention feature space, and $\tilde{\mathbf{v}}_t$ denotes the value update written into the state.

Channel-wise geometric retention. The scalar retention factor γ_t assigns a single lifetime to all evidence, which is insufficient for streaming geometry: local correspondences are short-lived, motion cues persist over moderate horizons, scene structure should survive across windows, and metric scale must remain stable over long sequences. We therefore replace γ_t with a channel-wise retention vector:

$$\boldsymbol{\gamma}_t = \sigma(\mathbf{W}_\gamma \mathbf{x}_t + \mathbf{b}_\gamma) \in (0,1)^d, \qquad \mathbf{S}_t = \mathrm{diag}(\boldsymbol{\gamma}_t)\mathbf{S}_{t-1} + \phi(\mathbf{k}_t)\tilde{\mathbf{v}}_t^\top. \tag{6}$$

Each channel c then has its own temporal influence factor and effective retention horizon:

$$K_{\mathrm{time}}^{(c)}(t,i) = \prod_{j=i+1}^{t} \gamma_j^{(c)}, \qquad \tau^{(c)} = -\frac{1}{\log \bar{\gamma}^{(c)}}. \tag{7}$$

Low-γ channels adapt rapidly to incoming observations, while high-γ channels preserve information over longer horizons. Together, they provide a learned spectrum of evidence lifetimes. Geometric cues can be represented across multiple retention bands, consistent with the distributed metric-related signal observed in our probing analysis in Appendix E.1.

Relation to TTT and linear attention. Eq. (6) admits an online-learning interpretation: the state adapts to incoming geometric evidence, similar to Test-Time Training (TTT). Explicit per-frame TTT optimization is costly for ultra-long streams, while TTT with KV binding admits an equivalent linear-attention form [25]. This links online adaptation to efficient recurrent attention and places our update within the family of gated linear attention mechanisms [18, 54, 62].

HorizonStream achieves this online recurrent form through a geometric state $\mathbf{S}_t$ and channel-wise retention $\boldsymbol{\gamma}_t$: $\mathbf{S}_t$ summarizes cross-window reconstruction evidence, while $\boldsymbol{\gamma}_t$ controls the temporal influence of each geometric channel. This yields an adaptive, efficient, and bounded recurrent update for long-range geometric propagation. Appendix A analyzes the long-sequence degradation of causal softmax attention and ungated recurrence.

3.3 Gated Window Attention

Geometric Linear Attention propagates compressed cross-window evidence, but accurate local reconstruction still requires fine-grained correspondences within each window. We instantiate the short-range spatial factor K_{spatial} with Gated Window Attention, which selects local evidence using image content and relative 3D layout before it enters the long-range state.

Head-wise output gating. To make the spatial kernel robust to sink-like concentration and noisy matches, we assign each attention head a reliability gate [29]. For head h,

$$g_h = \sigma(\mathbf{W}_g \bar{\mathbf{x}} + b_g), \qquad \tilde{\mathbf{y}}_h = g_h \cdot \mathbf{y}_h, \tag{8}$$

where $\bar{\mathbf{x}}$ is the mean-pooled window feature, $\mathbf{y}_h$ is the head output, σ is the sigmoid function, and $\mathbf{W}_g, b_g$ are learnable projection parameters. The gate downweights unreliable heads and preserves heads that support local matching.

Spatiotemporal RoPE. We extend RoPE [39] to three axes (time, height, and width) to encode relative spatiotemporal layout. For a patch at frame t and spatial location (y, x), we set $\pi = (t{+}1, y{+}1, x{+}1)$, split query and key vectors into three parts, and rotate each part along one axis. This makes attention depend on relative space-time offsets. We periodically reset the temporal index to avoid unbounded positional growth, while MRT and pose tokens use $\pi = (0, 0, 0)$. Together, gating controls head reliability and Spatiotemporal RoPE supplies relative geometric structure.

Metric Readout Tokens (MRT) and relative pose fusion. Long streaming reconstruction requires metric scale and pose to remain consistent across windows. Inspired by scale-token and metric-prediction designs [11, 19], MRT participates in Geometric Linear Attention and aggregates metric-related information from the recurrent state across the learned retention spectrum, extending metric readout from local context to sequence-level evidence.

Each frame includes a learned Metric Readout Token $\mathbf{z}^{\mathrm{metric}}$. A scale head predicts $\hat{s} = \exp\big(g(\mathbf{z}^{\mathrm{metric}})\big)$, which rescales translation and depth:

$$\hat{\mathbf{t}} = \hat{s} \cdot \hat{\mathbf{t}}^{\,\mathrm{raw}}, \qquad \hat{D} = \hat{s} \cdot \hat{D}^{\mathrm{raw}}. \tag{9}$$

For pose, we use relative pose fusion over pose tokens in the local window. A transformer head jointly attends to these tokens and estimates a consensus relative pose for the current frame with respect to the window context. This avoids relying on sequential keyframe chaining [11], where composition errors accumulate over long rollouts. Depth is produced by a DPT head with scale injection.

3.4 Architecture

Backbone. HorizonStream uses a ViT-L backbone initialized from VGGT [45] and DINOv2 [28]. Each frame contains image patch tokens, pose tokens, and a Metric Readout Token. The backbone alternates frame blocks and global blocks: frame blocks perform intra-frame self-attention, while global blocks adopt a hybrid temporal design that combines Gated Window Attention for dense intra-window tracking with Geometric Linear Attention layers interleaved at specific depths for cross-window memory updates.

Training objective. The model is supervised with pose, depth, and scale losses:

$$\mathcal{L} = \lambda_{pose}\,\mathcal{L}_{pose} + \lambda_{depth}\,\mathcal{L}_{depth} + \lambda_{scale}\,\mathcal{L}_{scale}. \tag{10}$$

Translation and depth are normalized by geometric scale factors. Depth loss is SmoothL1 with confidence weighting. Scale loss applies only on metric-scale samples.

Loop closure. To correct long-term accumulated drift during inference, an optional loop-closure module improves global revisit consistency. Inspired by VGGT-Long [12], we retrieve revisited frame pairs using stored compact global descriptors pooled from DINOv2 features. The retrieved candidates are re-fed into the network to estimate local geometric corrections. These are then converted into loop constraints to optimize the final global trajectory via pose graph optimization.

4 Experiments

4.1 Experimental Setup

Datasets. We evaluate on KITTI [15], vKITTI2 [3], Oxford Spires [41], ScanNet++ [56], TUM RGB-D [38], Waymo Open [40], VBR [2], ETH3D [33], and 7Scenes [35]. All sequences are evaluated at full length without subsampling. vKITTI2, 7Scenes, Waymo, and ScanNet++ are included in our training data; Waymo evaluation uses segments not seen during training. Detailed evaluation splits and per-dataset protocols are in Appendix D.

Baselines. We compare against three paradigms: (i) *optimization-based*: COLMAP [32], DPVO [43]/DPVO++, DROID-SLAM [42], MASt3R-SLAM [27], MASt3R-SfM [21], VGGT-SLAM [26]; (ii) *offline feed-forward*: VGGT-Long [12], FastVGGT [34], LoGeR [61] (and its optimization variant LoGeR*), Pi3-Chunk [49]; (iii) *online feed-forward*: CUT3R [46], TTT3R [6], STream3R [20], StreamVGGT [65], InfiniteVGGT [59], LongStream [11], Lingbot-map [5]. For CUT3R, TTT3R, and LoGeR, we report refresh and no-refresh variants to isolate the effect of periodic state reset. All baselines are evaluated on full sequences without subsampling using the released code and the default settings. Our inference code and evaluation scripts have been released.



Figure 4: **Qualitative comparison** on long-sequence 3D reconstruction. As sequence length grows, existing methods show pose degradation, drift, or collapse. Lingbot-map exhibits progressively stronger pose jitter over longer rollouts, while HorizonStream maintains stable pose estimation. For HorizonStream, dashed lines denote results without loop closure.

Table 1: Quantitative comparison on KITTI. We report mean ATE; "–" denotes OOM or repeated tracking failure, and LoGeR* denotes optimization-based LoGeR. Refresh/no-refresh variants degrade on long sequences, while trained with 48-frame batches, HorizonStream outperforms all streaming methods and approaches or surpasses offline methods with or without loop closure (LC).

	Methods	KITTI ATE ↓											
		00 4542 fr. 3.7 km	**01** 1101 fr. 2.5 km	**02** 4661 fr. 5.1 km	**03** 801 fr. 0.6 km	**04** 271 fr. 0.4 km	**05** 2761 fr. 2.2 km	**06** 1101 fr. 1.2 km	**07** 1101 fr. 0.7 km	**08** 4071 fr. 3.2 km	**09** 1591 fr. 1.7 km	**10** 1201 fr. 0.9 km	Avg.
Opt.-based	MASt3R-SLAM	–	530.37	–	18.87	88.98	159.43	92.00	–	263.75	–	153.07	186.64
	VGGT-SLAM	–	607.16	–	169.83	13.12	–	–	–	–	–	211.82	250.48
	COLMAP	139.12	3.83	71.99	1.46	112.77	20.37	10.95	7.80	21.72	21.19	4.52	37.79
	MASt3R-SfM	–	463.52	–	15.80	41.44	150.39	136.14	71.69	–	176.36	69.50	140.60
	DPVO	113.11	16.60	113.01	2.46	0.98	59.34	55.91	19.30	110.63	74.55	13.71	52.69
	DROID-SLAM	–	82.81	–	3.20	1.47	73.50	61.10	18.41	104.22	89.49	22.19	50.71
Offline Fwd.	VGGT-Long	8.64	61.21	52.72	8.78	4.20	9.88	4.67	2.66	72.98	31.84	27.71	25.94
	FastVGGT	–	705.39	–	62.38	10.27	157.74	124.43	69.27	–	190.10	194.75	189.29
	LoGeR	54.98	36.57	36.20	4.27	1.62	33.41	11.78	13.33	22.92	17.89	8.06	21.91
	LoGeR*	26.19	41.26	32.21	5.02	1.62	22.65	5.49	5.04	21.96	9.03	9.44	16.35
	LoGeR w/o refresh	166.05	631.14	226.65	66.09	4.55	125.16	98.32	12.38	203.24	127.28	185.19	167.82
Online Fwd.	CUT3R w/o refresh	185.89	651.52	296.98	148.06	22.17	155.61	132.54	77.03	238.39	205.94	193.39	209.78
	CUT3R w/ refresh	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 w/o refresh	190.93	546.84	218.77	105.28	11.62	153.12	132.94	70.95	180.57	211.01	133.00	177.73
	TTT3R w/ refresh	119.94	99.59	238.07	16.83	3.98	36.38	47.20	11.62	107.33	86.96	33.58	72.86
	STream3R	190.98	681.95	301.40	158.25	102.73	159.85	135.03	90.37	261.15	216.31	207.49	227.77
	StreamVGGT	191.93	653.06	303.35	157.50	108.24	160.46	133.71	89.00	263.95	216.69	209.80	226.15
	InfiniteVGGT	167.17	533.36	272.99	149.18	58.86	127.50	100.54	78.77	196.66	199.25	138.04	183.85
	LongStream	92.55	46.01	134.70	3.81	1.95	84.69	23.12	14.93	62.07	85.61	21.48	51.90
	Lingbot-map	30.80	64.74	82.29	**2.49**	0.85	16.55	6.27	8.92	39.32	**17.99**	**7.96**	25.29
	Ours	26.40	**20.62**	84.62	5.15	**0.62**	12.82	**4.59**	5.49	**19.49**	25.73	11.71	19.75
	Ours w/ LC	**13.91**	**20.62**	**69.43**	5.15	**0.62**	**6.86**	6.50	**2.67**	**19.49**	23.86	11.71	**16.44**

4.2 Implementation Details

Training mirrors streaming inference: both training stages use 48-frame samples, processed sequentially in 21-frame computational chunks, with the Geometric Linear Attention state propagating sequentially across chunks via a causal window. The pose prediction window is $W=10$, so short-term history spans 10 frames. Training proceeds in two stages: Stage 1 on 64 A800 GPUs for 60k iterations, Stage 2 on 64 H20 GPUs for 40k iterations with more long-sequence data. We use AdamW with learning rate 2×10^{-5} and cosine schedule with 2000 warmup steps. The reported GPU memory usage excludes the optional loop-closure module. Additional architecture specifications are in Appendix C.

Training data. We train on 24 datasets spanning indoor, outdoor driving, large-scale reconstruction, and synthetic environments, including ScanNet++ [56], Hypersim [31], Replica [37], 7Scenes [35], ARKitScenes [1], WildRGB-D [52], Waymo [40], vKITTI2 [3], Mapillary [50], MegaDepth [23], BlendedMVS [55], DL3DV [24], CO3Dv2 [30], TartanAir [48], PointOdyssey [63], OmniWorld [64], MatrixCity [22], and internal long-sequence data, among others. Training clips use temporal strides from 1 to 8. For unordered image sets, we build pseudo-temporal sequences by traversing the camera

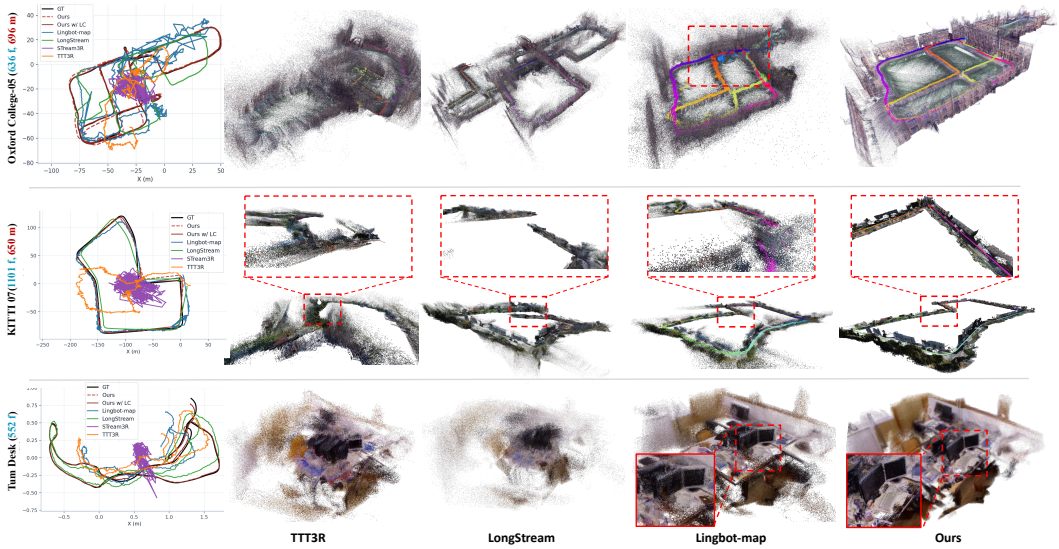


Figure 5: **Qualitative comparison** on 3D reconstruction. Left: trajectory. Right: 3D reconstruction. HorizonStream maintains stable geometry. Lingbot-map preserves trajectory direction but exhibits increasing jitter, causing point cloud overlap. For HorizonStream, dashed trajectory lines denote results without loop closure, and solid trajectory lines denote results with loop closure.

Table 2: **Quantitative comparison** across datasets. VKITTI2, Waymo and ScanNet++ are in-domain training datasets. HorizonStream performs strongly in both settings.

	Method	Calib.-free	ATE (m) ↓						FPS↑
			VKITTI2	KITTI	Oxford	ScanNet++	TUM	Waymo	
Opt.-based	MASt3R-SLAM	✓	81.55	186.64	37.73	0.47	0.08	7.63	7.40
	VGGT-SLAM	✓	19.23	250.48	31.00	0.29	0.12	7.43	15.80
	COLMAP	✓	9.59	37.79	15.57	GT	0.19	25.63	0.20
	MASt3R-SfM	✓	49.48	140.60	32.13	1.50	0.39	3.95	0.30
	DPVO++	✗	0.38	52.69	34.03	0.91	0.10	1.35	19.30
	DROID-SLAM	✗	1.12	50.71	31.08	0.97	0.11	6.67	13.60
Offline Fwd.	VGGT-Long	✓	0.91	25.94	21.90	0.13	0.08	1.78	4.80
	FastVGGT	✓	21.52	189.29	36.58	1.56	0.42	1.28	14.20
	LoGeR	✓	1.66	21.91	18.70	0.50	0.07	0.96	16.0
	LoGeR*	✓	2.45	16.35	15.79	0.43	0.08	0.55	9.1
Online Fwd.	CUT3R	✓	47.66	209.78	32.44	1.27	0.54	9.40	19.90
	TTT3R	✓	24.18	177.73	36.21	0.55	0.31	3.49	22.00
	STream3R	✓	68.96	227.77	37.57	1.75	0.63	42.20	8.20
	StreamVGGT	✓	68.51	226.15	37.25	1.70	0.63	45.10	19.10
	InfiniteVGGT	✓	58.63	183.85	31.82	1.66	0.21	20.56	5.30
	LongStream	✓	1.61	51.90	19.82	0.49	0.08	0.74	17.10
	Lingbot-map	✓	1.30	25.29	15.46	0.52	**0.04**	1.66	11.9
	Ours	✓	**0.94**	19.75	9.38	**0.40**	**0.04**	**0.46**	13.20
	Ours w/ LC	✓	**0.94**	**16.44**	**8.71**	**0.40**	**0.04**	**0.46**	10.45

graph. Frames are randomly permuted within each chunk with probability 0.2, while the cross-chunk order is preserved. Stage 1 focuses on short-window pose accuracy; Stage 2 incorporates longer source videos while retaining 48-frame training samples, increasing the coverage of long-sequence scenes and revisit patterns. Full list and per-stage sampling ratios are in Appendix C.3.

4.3 Camera Trajectory Estimation

Long-short sequence generalization. Tab. 1, 2, and 3 report mean ATE for trajectory estimation from indoor scenes to KITTI-scale driving and ultra-long VBR sequences exceeding 10,000 frames. On indoor benchmarks, HorizonStream is evaluated on the full sequences without downsampling. It achieves the best overall performance among online methods and remains competitive with offline approaches. As sequence length grows, existing streaming methods show pose degradation, severe jitter, or collapse; Lingbot-map can achieve competitive ATE, but its pose becomes increasingly jittery over longer sequences, as shown in Fig. 4. HorizonStream remains stable across all sequences.

KV-cache contamination. Refresh/no-refresh variants of CUT3R, TTT3R, and LoGeR isolate periodic state reset. Without refresh, all three degrade sharply, indicating temporal-state contamination rather than limited model capacity. HorizonStream avoids periodic refresh by discounting stale evidence and maintaining a bounded geometric state throughout the sequence.

Table 3: **Quantitative comparison** on VBR.

	Method	VBR ATE ↓							Avg.
		colosseo_0 8815 fr. 1.45 km	campus_0 12042 fr. 2.73 km	campus_1 11671 fr. 2.95 km	pincio_0 11142 fr. 1.27 km	spagna_0 14141 fr. 1.56 km	diag_0 10021 fr. 1.02 km	ciampino_1 18846 fr. 5.20 km	
Opt./Offline	VGGT-SLAM	101.00	93.51	71.74	66.42	57.00	33.64	124.10	78.20
	VGGT-Long w/o LC	81.54	118.59	98.21	53.44	46.92	30.80	170.30	85.69
	VGGT-Long	39.56	118.59	98.21	53.44	50.27	30.80	172.13	80.43
	LoGeR	31.77	27.90	30.80	17.96	21.33	32.25	34.16	28.02
	LoGeR*	55.32	13.27	16.79	9.18	18.32	29.45	34.32	25.24
	Pi3-Chunk	77.09	78.50	65.77	41.99	44.76	23.81	111.72	63.38
Online Fwd.	CUT3R	82.63	42.25	43.16	46.65	44.62	28.62	175.83	66.25
	TTT3R	75.52	59.44	56.55	33.87	37.33	**18.49**	173.71	64.99
	InfiniteVGGT	83.91	123.65	100.00	70.73	56.25	31.58	–	91.60
	LongStream	72.52	100.57	105.55	43.47	59.31	32.35	131.78	77.93
	Lingbot-map	16.70	23.61	10.37	29.37	24.29	24.12	64.24	27.53
	Ours	37.42	**22.46**	22.49	22.63	23.52	22.46	26.10	25.30
	Ours w/ LC	**12.76**	28.54	**8.49**	**17.24**	**23.06**	24.05	**17.76**	**18.84**

Table 4: **Quantitative comparison** of CD (↓) and F1 (↑) on multi-view reconstruction benchmarks.

	Method	ETH3D		Oxford Spires		7Scenes		TUM	
		CD ↓	F1@0.25 ↑	CD ↓	F1@4 ↑	CD ↓	F1@0.25 ↑	CD ↓	F1@0.25 ↑
Offline/Opt.	VGGT-Long	0.24	0.84	6.37	0.72	6.31	0.70	0.87	0.75
	MASt3R-SLAM	0.89	0.31	14.59	0.35	6.32	0.71	0.10	0.92
	VGGT-SLAM	0.78	0.72	11.51	0.32	6.37	0.71	0.10	0.93
	FastVGGT	0.50	0.70	7.97	0.63	5.99	0.69	0.07	0.94
	LoGeR	0.09	0.90	1.92	0.85	6.81	0.71	0.06	0.96
Online	StreamVGGT	1.86	0.14	15.45	0.27	6.23	0.66	0.39	0.59
	STream3R	1.81	0.14	15.44	0.26	6.31	0.72	0.15	0.86
	CUT3R	0.41	0.60	8.22	0.41	6.35	0.48	1.51	0.32
	TTT3R	0.43	0.59	9.95	0.30	6.63	0.48	0.86	0.29
	InfiniteVGGT	0.46	0.61	9.65	0.43	6.43	0.69	0.22	0.81
	LongStream	0.77	0.55	6.28	0.55	**2.26**	0.64	0.23	0.67
	Lingbot-map	0.37	0.68	8.69	0.43	6.33	0.72	**0.08**	0.94
	Ours	**0.32**	**0.74**	**4.97**	**0.89**	2.98	**0.93**	**0.08**	**0.95**

Table 5: Video depth estimation results on KITTI.

Method	Abs Rel ↓	$\delta < 1.25$ ↑
DUSt3R-GA	0.144	81.3
MASt3R-GA	0.183	74.5
MonST3R-GA	0.168	74.4
VGGT	0.061	**97.0**
Spann3R	0.198	73.7
CUT3R	0.118	88.1
Point3R	0.136	84.2
StreamVGGT	0.173	72.1
STream3R	0.080	94.7
InfiniteVGGT	0.170	78.6
LoGeR	0.090	93.0
LongStream	0.120	87.0
Lingbot-map	0.098	90.7
Ours	**0.057**	94.8

4.4 Dense Reconstruction and Depth

Tab. 4 and Tab. 5 report reconstruction and depth accuracy. Note that 7Scenes is part of our training data. HorizonStream achieves the best online reconstruction quality across four benchmarks, mainly due to more accurate pose estimation. On 7Scenes, several baselines have inflated mean CD due to large errors on Chess, Pumpkin, and RedKitchen. On KITTI depth, HorizonStream approaches the best offline methods among the compared baselines.

Ablation study. *Geometric Linear Attention.* Removing it entirely causes severe drift, confirming the necessity of long-term state. Disabling channel-wise gating or replacing it with TTT-like fast weights both degrade performance, especially at longer horizons, showing that per-channel bounded retention is critical. Fig. 6a visualizes the learned effective lifetimes $\tau = -1/\log\bar{\gamma}$, which form a continuous spectrum across channels and layers. Fig. 6b further shows that replacing this learned spectrum with any fixed band degrades accuracy, confirming the necessity of multi-timescale retention.

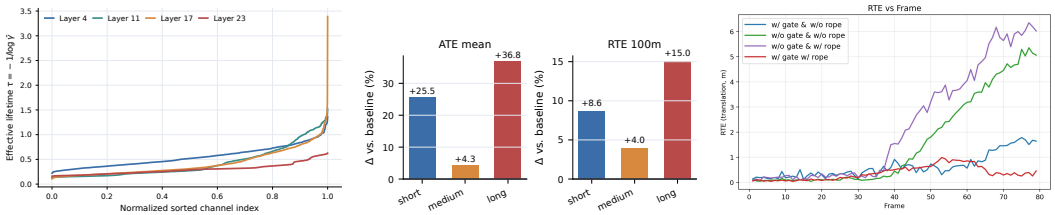
Table 6: ATE (↓) ablation on vKITTI2.

Variant	80f	200f	400f
Full model	**0.42**	**0.71**	**1.20**
Geometric Linear Attention			
w/o Geometric Linear Attention	0.83	2.06	5.38
w/o channel-wise gating	0.67	1.43	3.21
replace with TTT-like fast weight	0.58	1.56	3.96
Gated Window Attention			
w/o Gated Window Attention	0.78	2.64	7.46
w/o head-wise output gating	0.61	1.74	4.06
w/o Geometric RoPE, 2D spatial only	0.64	1.22	2.58
Scale and pose			
w/o MRT	0.55	1.32	3.34
single-token pose, no aggregation	0.51	1.10	2.67

Gated Window Attention. Removing it yields the severe degradation, reflecting the importance of fine-grained spatial matching within each window. Fig. 6c shows that head-wise output gating and Spatiotemporal RoPE are complementary: removing either substantially increases drift over long sequences.

Scale and pose readout. Metric Readout Tokens and multi-token pose aggregation each contribute consistent gains. Additional results on loop closure, memory and runtime scaling, and training convergence are in Appendix E.

Discussion. HorizonStream predicts poses using a local window of only 10 frames, suggesting that compact local geometric evidence is sufficient for accurate pose estimation while reducing memory cost and improving inference speed. A larger pose window may further improve the model's internal loop-closure ability. Additionally, for extremely long sequences with repeated revisits, the fixed-size



(a) (b) (c)

Figure 6: **(a) Learned retention spectra in Geometric Linear Attention.** Effective lifetimes $\tau = -1/\log \bar{\gamma}$ vary across channels and layers. Layer 4 exhibits broad mid-range retention, while Layer 11 develops a pronounced long-retention tail, supporting channel-wise multi-timescale propagation. **(b) Retention-band ablation.** Replacing the learned channel-wise retention spectrum with fixed short-, medium-, or long-horizon bands increases trajectory error, showing that stable long-sequence propagation requires learned multi-timescale retention. **(c) Long-sequence stability of Gated Window Attention.** Head-wise gating and 3D RoPE are complementary: removing either causes error growth over time, while using both keeps the model stable.

recurrent state still misses fine-grained details, as shown in Fig. 10. Dynamic foreground objects can also corrupt local geometric evidence in the input video. The optional loop-closure module is currently parameterized separately, and its optimization settings could be further refined. The method also relies on strong pretraining and substantial training compute, can be less reliable in texture-poor scenes, and may propagate pose or reconstruction errors to downstream robotic or autonomous-driving systems.

5 Conclusion

We presented HorizonStream, a streaming 3D reconstruction framework built on an evidence influence kernel that unifies long-term temporal memory and short-term spatial matching. Its channel-wise retention, Gated Window Attention, and Metric Readout Tokens jointly support multi-timescale evidence propagation, reliable local matching, and consistent metric readout. Trained on 48-frame clips, it generalizes to sequences exceeding 10,000 frames with constant memory and linear time. Experiments across diverse indoor and outdoor datasets demonstrate strong trajectory accuracy and coherent dense reconstruction, highlighting the importance of learned evidence lifetimes for stable long-horizon streaming.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (No. 62406267), Guangdong Provincial Project (No. 2024QN11X072), Guangzhou-HKUST(GZ) Joint Funding Program (No. 2025A03J3956) and Guangzhou Municipal Education Project (No. 2024312122).

References

[1] Gilad Baruch, Zhuoyuan Chen, Afshin Dehghan, Tal Dimry, Yuri Feigin, Peter Fu, Thomas Gebauer, Brandon Joffe, Daniel Kurz, Arik Schwartz, et al. Arkitscenes: A diverse real-world dataset for 3d indoor scene understanding using mobile rgb-d data. *arXiv preprint arXiv:2111.08897*, 2021.

[2] Leonardo Brizi, Emanuele Giacomini, Luca Di Giammarino, Simone Ferrari, Omar Salem, Lorenzo De Rebotti, and Giorgio Grisetti. VBR: A vision benchmark in rome. *arXiv preprint arXiv:2404.11322*, 2024.

[3] Yohann Cabon, Naila Murray, and Martin Humenberger. Virtual KITTI 2. *arXiv preprint arXiv:2001.10773*, 2020.

[4] Carlos Campos, Richard Elvira, Juan J. Gomez Rodriguez, Jose M. M. Montiel, and Juan D. Tardos. ORB-SLAM3: An accurate open-source library for visual, visual-inertial and multi-map SLAM. *IEEE Transactions on Robotics*, 37(6), 2021.

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

[6] Xingyu Chen, Yue Chen, Yuliang Xiu, Andreas Geiger, and Anpei Chen. TTT3R: 3D reconstruction as test-time training. *arXiv preprint arXiv:2509.26645*, 2025.

[7] Chong Cheng, Yu Hu, Sicheng Yu, Beizhen Zhao, Zijian Wang, and Hao Wang. Reggs: Unposed sparse views gaussian splatting with 3dgs registration, 2025. URL https://arxiv.org/abs/2507.08136.

[8] Chong Cheng, Gaochao Song, Yiyang Yao, Qinzheng Zhou, Gangjian Zhang, and Hao Wang. Graph-guided scene reconstruction from images with 3d gaussian splatting, 2025. URL https://arxiv.org/abs/2502.17377.

[9] Chong Cheng, Zijian Wang, Sicheng Yu, Yu Hu, Nanjie Yao, and Hao Wang. Unposed 3dgs reconstruction with probabilistic procrustes mapping, 2025. URL https://arxiv.org/abs/2507.18541.

[10] Chong Cheng, Sicheng Yu, Zijian Wang, Yifan Zhou, and Hao Wang. Outdoor monocular slam with global scale-consistent 3d gaussian pointmaps, 2025. URL https://arxiv.org/abs/2507.03737.

[11] Chong Cheng, Xianda Chen, Tao Xie, Wei Yin, Weiqiang Ren, Qian Zhang, Xiaoyang Guo, and Hao Wang. LongStream: Long-sequence streaming autoregressive visual geometry. In *CVPR*, 2026.

[12] Kai Deng, Zexin Ti, Jiawei Xu, Jian Yang, and Jin Xie. Vggt-long: Chunk it, loop it, align it – pushing vggt's limits on kilometer-scale long rgb sequences, 2025. URL https://arxiv.org/abs/2507.16443.

[13] Xianze Fang, Jingnan Gao, Zhe Wang, Zhuo Chen, Xingyu Ren, Jiangjing Lyu, Qiaomu Ren, Zhonglei Yang, Xiaokang Yang, Yichao Yan, and Chengfei Lyu. Dens3R: A foundation model for 3D geometry prediction. In *The Fourteenth International Conference on Learning Representations*, 2026.

[14] Jingnan Gao, Zhe Wang, Xianze Fang, Xingyu Ren, Zhuo Chen, Shengqi Liu, Yuhao Cheng, Jiangjing Lyu, Xiaokang Yang, and Yichao Yan. MoRE: 3D visual geometry reconstruction meets mixture-of-experts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026.

[15] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? The KITTI Vision Benchmark Suite. In *CVPR*, 2012.

[16] Xiangming Gu, Tianyu Pang, Chao Du, Qian Liu, Fengzhuo Zhang, Cunxiao Du, Ye Wang, and Min Lin. When attention sink emerges in language models: An empirical view. In *ICLR*, 2025.

[17] Yu Hu, Chong Cheng, Sicheng Yu, Xiaoyang Guo, and Hao Wang. Vggt4d: Mining motion cues in visual geometry transformers for 4d scene reconstruction, 2025. URL https://arxiv.org/abs/2511.19971.

[18] Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and Francois Fleuret. Transformers are RNNs: Fast autoregressive transformers with linear attention. In *ICML*, 2020.

[19] Nikhil Keetha, Norman Müller, Johannes Schönberger, Lorenzo Porzi, Yuchen Zhang, Tobias Fischer, Arno Knapitsch, Duncan Zauss, Ethan Weber, Nelson Antunes, Jonathon Luiten, Manuel Lopez-Antequera, Samuel Rota Bulò, Christian Richardt, Deva Ramanan, Sebastian Scherer, and Peter Kontschieder. Mapanything: Universal feed-forward metric 3d reconstruction, 2025. URL https://arxiv.org/abs/2509.13414.

[20] Yushi Lan, Yihang Luo, Fangzhou Hong, Shangchen Zhou, Honghua Chen, Zhaoyang Lyu, Shuai Yang, Bo Dai, Chen Change Loy, and Xingang Pan. Stream3r: Scalable sequential 3d reconstruction with causal transformer, 2025. URL https://arxiv.org/abs/2508.10893.

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

[22] Yixuan Li, Lihan Jiang, Linning Xu, Yuanbo Xiangli, Zhenzhi Wang, Dahua Lin, and Bo Dai. MatrixCity: A large-scale city dataset for city-scale neural rendering and beyond. In *ICCV*, 2023.

[23] Zhengqi Li and Noah Snavely. MegaDepth: Learning single-view depth prediction from internet photos. In *CVPR*, 2018.

[24] Lu Ling, Yichen Sheng, Zhi Tu, Wentian Zhao, Cheng Xin, Kun Wan, Lantao Yu, Qianyu Guo, Zixun Yu, Yawen Lu, et al. DL3DV-10K: A large-scale scene dataset for deep learning-based 3D vision. In *CVPR*, 2024.

[25] Junchen Liu, Sven Elflein, Or Litany, Zan Gojcic, and Ruilong Li. Test-time training with kv binding is secretly linear attention, 2026. URL `https://arxiv.org/abs/2602.21204`.

[26] Dominic Maggio, Hyungtae Lim, and Luca Carlone. VGGT-SLAM: Dense RGB SLAM optimized on the SL(4) manifold. *arXiv preprint arXiv:2505.12549*, 2025.

[27] Riku Murai, Eric Dexheimer, and Andrew J. Davison. MASt3R-SLAM: Real-time dense SLAM with 3D reconstruction priors. In *CVPR*, 2025.

[28] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2024. URL `https://arxiv.org/abs/2304.07193`.

[29] Zihan Qiu, Zekun Wang, Bo Zheng, Zeyu Huang, Kaiyue Wen, Songlin Yang, Rui Men, Le Yu, Fei Huang, Suozhi Huang, et al. Gated attention for large language models: Non-linearity, sparsity, and attention-sink-free. *arXiv preprint arXiv:2505.06708*, 2025.

[30] Jeremy Reizenstein, Roman Shapovalov, Philipp Henzler, Luca Sbordone, Patrick Labatut, and David Novotny. Common objects in 3D: Large-scale learning and evaluation of real-life 3D category reconstruction. In *ICCV*, 2021.

[31] Mike Roberts, Jason Ramapuram, Anurag Ranjan, Atulit Kumar, Miguel Angel Bautista, Nathan Paczan, Russ Webb, and Joshua M. Susskind. Hypersim: A photorealistic synthetic dataset for holistic indoor scene understanding. In *ICCV*, 2021.

[32] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, 2016.

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

[34] You Shen, Zhipeng Zhang, Yansong Qu, and Liujuan Cao. FastVGGT: Training-free acceleration of visual geometry transformer. *arXiv preprint arXiv:2509.02560*, 2025.

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

[36] Gaochao Song, Chong Cheng, and Hao Wang. Gvkf: Gaussian voxel kernel functions for highly efficient surface reconstruction in open scenes, 2024. URL `https://arxiv.org/abs/2411.01853`.

[37] Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J. Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, et al. The Replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019.

[38] Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. A benchmark for the evaluation of RGB-D SLAM systems. In *IROS*, 2012.

[39] Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. RoFormer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568, 2024.

[40] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo Open Dataset. In *CVPR*, 2020.

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

[42] Zachary Teed and Jia Deng. DROID-SLAM: Deep visual SLAM for monocular, stereo, and RGB-D cameras. In *NeurIPS*, 2021.

[43] Zachary Teed, Lahav Lipson, and Jia Deng. Deep patch visual odometry. In *NeurIPS*, 2023.

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

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

[46] Qianqian Wang, Yifei Zhang, Aleksander Holynski, Alexei A Efros, and Angjoo Kanazawa. Continuous 3d perception model with persistent state. *arXiv preprint arXiv:2501.12387*, 2025.

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

[48] Wenshan Wang, Delong Zhu, Xiangwei Wang, Yaoyu Hu, Yuheng Qiu, Chen Wang, Yafei Hu, Ashish Kapoor, and Sebastian Scherer. TartanAir: A dataset to push the limits of visual SLAM. In *IROS*, 2020.

[49] Yifan Wang, Jianjun Zhou, Haoyi Zhu, Wenzheng Chang, Yang Zhou, Zizun Li, Junyi Chen, Jiangmiao Pang, Chunhua Shen, and Tong He. π^3: Permutation-equivariant visual geometry learning, 2025. URL `https://arxiv.org/abs/2507.13347`.

[50] Frederik Warburg, Søren Hauberg, Manuel Lopez-Antequera, Pau Gargallo, Yubin Kuang, and Javier Civera. Mapillary street-level sequences: A dataset for lifelong place recognition. In *CVPR*, 2020.

[51] Yuqi Wu, Wenzhao Zheng, Jie Zhou, and Jiwen Lu. Point3R: Streaming 3D reconstruction with explicit spatial pointer memory. *arXiv preprint arXiv:2507.02863*, 2025.

[52] Hongchi Xia, Yang Fu, Sifei Liu, and Xiaolong Wang. RGBD objects in the wild: Scaling real-world 3D object learning from RGB-D videos. In *CVPR*, 2024.

[53] Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. In *ICLR*, 2024.

[54] Songlin Yang, Jan Kautz, and Ali Hatamizadeh. Gated delta networks: Improving mamba2 with delta rule, 2025. URL `https://arxiv.org/abs/2412.06464`.

[55] Yao Yao, Zixin Luo, Shiwei Li, Jingyang Zhang, Yufan Ren, Lei Zhou, Tian Fang, and Long Quan. BlendedMVS: A large-scale dataset for generalized multi-view stereo networks. In *CVPR*, 2020.

[56] Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. ScanNet++: A high-fidelity dataset of 3D indoor scenes. In *ICCV*, 2023.

[57] Sicheng Yu, Chong Cheng, Yifan Zhou, Xiaojun Yang, and Hao Wang. Rgb-only gaussian splatting slam for unbounded outdoor scenes, 2025. URL `https://arxiv.org/abs/2502.15633`.

[58] Zhu Yu, Jingnan Gao, Runmin Zhang, Lingteng Qiu, Zhengyi Zhao, Rui Peng, Yichao Yan, Kejie Qiu, Siyu Zhu, Zilong Dong, Si-Yuan Cao, and Hui-Liang Shen. Towards consistent video geometry estimation, 2026. URL https://arxiv.org/abs/2605.30060.

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

[60] Junyi Zhang, Charles Herrmann, Junhwa Hur, Varun Jampani, Trevor Darrell, Forrester Cole, Deqing Sun, and Ming-Hsuan Yang. MonST3R: A simple approach for estimating geometry in the presence of motion. In *ICLR*, 2025.

[61] Junyi Zhang, Charles Herrmann, Junhwa Hur, Chen Sun, Ming-Hsuan Yang, Forrester Cole, Trevor Darrell, and Deqing Sun. Loger: Long-context geometric reconstruction with hybrid memory, 2026. URL https://arxiv.org/abs/2603.03269.

[62] Yu Zhang, Zongyu Lin, Xingcheng Yao, Jiaxi Hu, Fanqing Meng, et al. Kimi linear: An expressive, efficient attention architecture. *arXiv preprint arXiv:2510.26692*, 2025.

[63] Yang Zheng, Adam W. Harley, Bokui Shen, Gordon Wetzstein, and Leonidas J. Guibas. PointOdyssey: A large-scale synthetic dataset for long-term point tracking. In *ICCV*, 2023.

[64] Yang Zhou, Yifan Wang, Jianjun Zhou, Wenzheng Chang, Haoyu Guo, Zizun Li, Kaijing Ma, Xinyue Li, Yating Wang, Haoyi Zhu, Mingyu Liu, Dingning Liu, Jiange Yang, Zhoujie Fu, Junyi Chen, Chunhua Shen, Jiangmiao Pang, Kaipeng Zhang, and Tong He. Omniworld: A multi-domain and multi-modal dataset for 4d world modeling, 2025. URL https://arxiv.org/abs/2509.12201.

[65] Dong Zhuo, Wenzhao Zheng, Jiahe Guo, Yuqi Wu, Jie Zhou, and Jiwen Lu. Streaming 4d visual geometry transformer, 2026. URL https://arxiv.org/abs/2507.11539.

A Geometric Attention Dilution

We formalize why causal softmax attention cannot serve as long-range cross-window memory in streaming 3D reconstruction.

Let Ω_t denote the set of 3D points visible at time t, and define the *geometric relevance* of frame i to frame t as the co-visibility ratio $r(i,t) = |\Omega_i \cap \Omega_t|/|\Omega_t|$. For a camera exploring new regions, the co-visibility set $\mathcal{R}_t = \{i \le t : r(i,t) > 0\}$ is bounded by $|\mathcal{R}_t| \le W_{\text{geo}}$, determined by scene geometry and camera speed.

Proposition 1 (Geometric Attention Dilution)**.** *Let $\alpha_i = \mathrm{softmax}(\mathbf{q}_t^\top \mathbf{k}_i/\sqrt{d})_{i=1}^t$ be causal softmax attention weights with bounded scores $|\mathbf{q}_t^\top \mathbf{k}_i/\sqrt{d}| \le M$. The total attention on geometrically relevant frames satisfies:*

$$\sum_{i\in\mathcal{R}_t} \alpha_i \le \frac{1}{1 + \dfrac{t - W_{\text{geo}}}{W_{\text{geo}}} \cdot e^{-2M}}. \tag{11}$$

For $t > W_{\text{geo}}(1 + e^{2M})$, more than half the attention mass falls on geometrically irrelevant frames. Even under perfect score discrimination, the relevant fraction decays as $O(W_{\text{geo}} e^{2M}/t)$.

Proof. Assign the best-case scores: $+M$ to all W_{geo} relevant frames, $-M$ to all others. Then:

$$\sum_{i\in\mathcal{R}_t} \alpha_i \le \frac{W_{\text{geo}} \cdot e^{M}}{W_{\text{geo}} \cdot e^{M} + (t - W_{\text{geo}}) \cdot e^{-M}}.$$

Dividing numerator and denominator by $W_{\text{geo}} \cdot e^M$ yields (11). Any suboptimal score assignment only worsens the bound. For $t \gg W_{\text{geo}} e^{2M}$, the bound is $O(W_{\text{geo}} e^{2M}/t) \to 0$. □

Remark 1. *In practice, models mitigate the wasted attention by concentrating irrelevant mass onto* sink tokens *[53, 16] whose values collapse toward zero. This symptom does not resolve the underlying problem: the geometrically useful signal fraction still vanishes as $O(1/t)$, and the $O(t)$ KV-cache cost remains. Both properties rule out causal softmax as a cross-window memory mechanism.*

B Extended Theoretical Analysis

B.1 Zero-Forgetting Contamination and Stability

We state and prove the two core propositions motivating selective forgetting in the recurrent geometric state.

Proposition 2 (Zero-Forgetting Contamination)**.** *Under zero forgetting ($\gamma \equiv 1$), the initial state $\mathbf{S}_0$ contributes to every output with undiminished magnitude:*

$$\mathbf{o}_t = \mathbf{q}_t^\top \mathbf{S}_0 + \sum_{i=1}^{t} \mathbf{q}_t^\top \mathbf{k}_i \tilde{\mathbf{v}}_i^\top.$$

No amount of new evidence can dilute the initial state.

Proof. Under the ungated update $\mathbf{S}_t = \mathbf{S}_{t-1} + \mathbf{k}_t \tilde{\mathbf{v}}_t^\top$, unrolling gives $\mathbf{S}_t = \mathbf{S}_0 + \sum_{i=1}^t \mathbf{k}_i \tilde{\mathbf{v}}_i^\top$. Hence $\mathbf{o}_t = \mathbf{q}_t^\top \mathbf{S}_t = \mathbf{q}_t^\top \mathbf{S}_0 + \sum_{i=1}^t \mathbf{q}_t^\top \mathbf{k}_i \tilde{\mathbf{v}}_i^\top$. The $\mathbf{q}_t^\top \mathbf{S}_0$ term is independent of t and never diminishes. □

Remark 2. *This is the root cause of the degradation observed in TTT without reset [6, 61], CUT3R [46], and standard linear attention when applied to long streaming sequences: the state is permanently anchored to initialization regardless of camera motion.*

Proposition 3 (Bounded Initial-State Influence)**.** *Under the channel-wise retention update* (6)*, if $\bar{\gamma} = \sup_{t,c} |\gamma_t^{(c)}| < 1$, the contribution of the initial state decays exponentially:*

$$\left\| \mathbf{q}_t^\top \left(\prod_{j=1}^{t} \mathrm{diag}(\boldsymbol{\gamma}_j) \right) \mathbf{S}_0 \right\| \le \|\mathbf{q}_t\| \cdot \|\mathbf{S}_0\|_F \cdot \bar{\gamma}^{\,t} \to 0 \quad \textit{as } t \to \infty.$$

Proof. Unrolling the gated recurrence gives

$$\mathbf{S}_t = \left(\prod_{j=1}^{t} \operatorname{diag}(\boldsymbol{\gamma}_j)\right) \mathbf{S}_0 + \sum_{i=1}^{t} \left(\prod_{j=i+1}^{t} \operatorname{diag}(\boldsymbol{\gamma}_j)\right) \mathbf{k}_i \tilde{\mathbf{v}}_i^\top.$$

Since $\| \operatorname{diag}(\boldsymbol{\gamma}_j)\|_{\mathrm{op}} \le \bar{\gamma}$, submultiplicativity gives

$$\left\| \prod_{j=1}^{t} \operatorname{diag}(\boldsymbol{\gamma}_j) \right\|_{\mathrm{op}} \le \bar{\gamma}^{\,t}.$$

Applying Cauchy–Schwarz yields the stated bound. □

Thus, channel-wise retention with $\bar{\gamma} < 1$ is a sufficient condition for closing the influence of the initial state. The recurrent state remains adaptive to incoming evidence rather than being anchored to its initialization.

B.2 Effective Memory Horizon

Proposition 4 (Per-Channel Memory Horizon)**.** *For a channel c with constant gate $\gamma^{(c)} \in (0, 1)$, define the effective memory horizon as $\tau^{(c)} = -1/\log \gamma^{(c)}$. Then the contribution of observation i to the output at time t decays as:*

$$w^{(c)}(t, i) = (\gamma^{(c)})^{t-i} = e^{-(t-i)/\tau^{(c)}}.$$

For $t - i > 3\tau^{(c)}$, the contribution is below 5% of its original weight.

Proof. Direct computation: $w^{(c)}(t, i) = (\gamma^{(c)})^{t-i} = e^{(t-i)\log \gamma^{(c)}} = e^{-(t-i)/\tau^{(c)}}$. At $t - i = 3\tau^{(c)}$: $w = e^{-3} \approx 0.050$. □

Corollary 1 (Heterogeneous Memory Partitioning)**.** *With channel-wise gating, the state $\mathbf{S}_t \in \mathbb{R}^{d\times d}$ is implicitly partitioned into subspaces with different memory lifetimes. Let $\mathcal{C}_{fast} = \{c : \gamma^{(c)} < \gamma_{th}\}$ and $\mathcal{C}_{slow} = \{c : \gamma^{(c)} \ge \gamma_{th}\}$ for some threshold γ_{th}. Then the fast subspace $\mathbf{S}_t[\mathcal{C}_{fast}, :]$ acts as a short-term feature buffer with horizon $\tau_{fast} \ll T$, while the slow subspace $\mathbf{S}_t[\mathcal{C}_{slow}, :]$ acts as a long-term geometric memory with horizon $\tau_{slow} \gg W$. This partitioning is learned end-to-end and adapts to the geometric content of the training data.*

B.3 State Norm Boundedness

A key practical concern is whether the persistent state $\mathbf{S}_t$ remains bounded as $t \to \infty$.

Proposition 5 (Bounded State Norm)**.** *Under the gated update $\mathbf{S}_t = \operatorname{diag}(\boldsymbol{\gamma}_t)\mathbf{S}_{t-1} + \mathbf{k}_t\tilde{\mathbf{v}}_t^\top$ with $\bar{\gamma} = \sup_{t,c} |\gamma_t^{(c)}| < 1$ and bounded inputs $\|\mathbf{k}_t\| \le B_k$, $\|\tilde{\mathbf{v}}_t\| \le B_v$ for all t, the Frobenius norm of the state is uniformly bounded:*

$$\|\mathbf{S}_t\|_F \le \bar{\gamma}^t \|\mathbf{S}_0\|_F + \frac{B_k B_v}{1 - \bar{\gamma}}.$$

In particular, $\limsup_{t\to\infty} \|\mathbf{S}_t\|_F \le B_k B_v/(1 - \bar{\gamma})$.

Proof. By submultiplicativity and the triangle inequality:

$$\begin{aligned} \|\mathbf{S}_t\|_F &\le \|\operatorname{diag}(\boldsymbol{\gamma}_t)\|_{\mathrm{op}} \|\mathbf{S}_{t-1}\|_F + \|\mathbf{k}_t\tilde{\mathbf{v}}_t^\top\|_F \\ &\le \bar{\gamma}\|\mathbf{S}_{t-1}\|_F + B_k B_v. \end{aligned}$$

Unrolling the recurrence: $\|\mathbf{S}_t\|_F \le \bar{\gamma}^t\|\mathbf{S}_0\|_F + B_k B_v \sum_{i=0}^{t-1} \bar{\gamma}^i \le \bar{\gamma}^t\|\mathbf{S}_0\|_F + B_k B_v/(1 - \bar{\gamma})$. □

Remark 3. *The bound in Proposition 5 guarantees numerical stability without periodic state resets. In contrast, ungated linear attention ($\bar{\gamma} = 1$) yields $\|\mathbf{S}_t\|_F \le \|\mathbf{S}_0\|_F + t \cdot B_k B_v$, which grows linearly and eventually requires resets to prevent overflow, as observed in TTT-based methods [61].*

B.4 Formal Connection to Test-Time Training

We formalize the relationship between Geometric Linear Attention and TTT.

Proposition 6 (Geometric Linear Attention as Discounted TTT)**.** *Consider a linear model* $f_{\mathbf{S}}(\mathbf{k}) = \mathbf{S}^\top \mathbf{k}$ *trained online to minimize* $\ell_t = \|\mathbf{S}^\top \mathbf{k}_t - \mathbf{v}_t\|^2$ *with the discounted objective* (2)*. One step of gradient descent at learning rate* η *on the discounted objective, starting from the previous iterate* $\mathbf{S}_{t-1}$ *discounted by* γ_t*, produces:*

$$\mathbf{S}_t = \gamma_t \mathbf{S}_{t-1} - \frac{\eta}{2} \nabla_{\mathbf{S}} \ell_t \big|_{\mathbf{S}=\gamma_t \mathbf{S}_{t-1}} = \gamma_t \mathbf{S}_{t-1} + \eta \mathbf{k}_t (\mathbf{v}_t - \gamma_t \mathbf{S}_{t-1}^\top \mathbf{k}_t)^\top.$$

When $\gamma_t \equiv 1$*, this reduces to the standard online linear regression update, which Liu et al. [25] showed is equivalent to linear attention. The gated form thus extends the TTT–linear-attention equivalence to the discounted setting:* Geometric Linear Attention is equivalent to discounted test-time training.

Proof. The gradient of ℓ_t at $\mathbf{S}' = \gamma_t \mathbf{S}_{t-1}$ is $\nabla_{\mathbf{S}} \ell_t|_{\mathbf{S}'} = 2\mathbf{k}_t(\mathbf{S}'^\top \mathbf{k}_t - \mathbf{v}_t)^\top$. Gradient descent: $\mathbf{S}_t = \mathbf{S}' - (\eta/2)\nabla \ell_t|_{\mathbf{S}'} = \gamma_t \mathbf{S}_{t-1} + \eta \mathbf{k}_t (\mathbf{v}_t - \gamma_t \mathbf{S}_{t-1}^\top \mathbf{k}_t)^\top$. Setting $\gamma_t = 1$ recovers $\mathbf{S}_t = \mathbf{S}_{t-1} + \eta \mathbf{k}_t (\mathbf{v}_t - \mathbf{S}_{t-1}^\top \mathbf{k}_t)^\top$, the undiscounted TTT/linear-attention update of Liu et al. [25]. □

C Implementation Details

C.1 Architecture Details

The backbone consists of 24 transformer layers alternating between frame blocks and global blocks. Geometric Linear Attention layers are placed at layers 4, 11, 17, and 23. Each Geometric Linear Attention layer reads and updates the persistent state after Gated Window Attention operates. The head-wise gate bias is initialized to 2.0 to preserve pretrained attention at the start of training. Geometric Linear Attention gates are initialized with high bias to produce $\gamma \approx 1$, gradually learning channel-wise retention as training progresses.

The pose consensus head uses a lightweight transformer with residual corrections over K rounds. Each round refines the translation, rotation quaternion, and focal length. The depth head uses DPT-style multi-scale fusion from four intermediate layers.

C.2 Training Hyperparameters

Input and model dimensions. Input images are resized to 518×518. The Geometric Linear Attention state has dimension $\mathbf{S} \in \mathbb{R}^{d \times d}$ with d=1024.

Scale loss is applied only on metric-scale samples. Depth loss uses SmoothL1 with confidence weighting. We apply random color jitter, random cropping, and random horizontal flip.

Local-window sampling. During training, we randomly vary the local-window size while keeping each training sample at 48 frames. The recurrent state is propagated across computational chunks. At inference, we use a local window of 10 frames as an accuracy–efficiency trade-off.

C.3 Training Data

We train on 24 datasets covering indoor, outdoor, driving, and synthetic environments. Video data is sampled with variable temporal stride from 1 to 8. Unordered image collections are converted to pseudo-temporal sequences via camera graph traversal. Tab. 7 lists per-dataset sampling ratios.

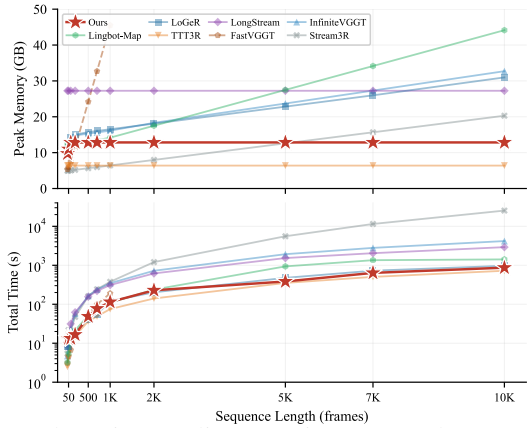


Figure 7: GPU memory and runtime scaling. HorizonStream keeps peak GPU memory nearly constant and scales smoothly to 10K frames, while competing methods require increasing GPU memory or higher runtime on long sequences.

Table 7: Training data composition. We use dataset-specific sampling ratios in two stages. Stage 1 emphasizes data diversity, while Stage 2 increases the proportion of metric-scale datasets.

Dataset	Stage 1 Ratio	Stage 2 Ratio	Metric Scale
blendedmvs/train	4.90%	2.40%	✗
megadepth	1.00%	–	✗
hypersim/train	7.30%	5.10%	✓
hypersim/val	1.50%	1.40%	✓
ase	9.80%	9.50%	✓
scannetpp	7.30%	6.10%	✓
tartanair	7.30%	6.10%	✓
vkitti2	9.80%	9.50%	✓
mapillary	9.80%	9.50%	✓
waymo	9.80%	9.50%	✓
wildrgbd/train	7.30%	7.10%	✓
co3dv2/train	7.00%	–	✗
dl3dv	9.80%	9.50%	✗
mapfree/train	4.90%	4.70%	✓
replica_niceslam	0.50%	0.86%	✗
7scenes	0.50%	0.86%	✗
GTAV_1080	1.00%	0.90%	✗
spring	0.50%	–	✗
point_odyssey/train	–	0.86%	✗
point_odyssey/val	–	0.86%	✗
ARKitscenes	–	0.86%	✓
unrealstereo4k	–	0.90%	✗
OmniWorld	–	4.70%	✓
matrixcity_d2 aerial	–	2.40%	✓
matrixcity_d2 street	–	2.40%	✓
Internal Long-Sequence Data	–	4.00%	✓

D Evaluation Dataset Details

We evaluate all sequences at full length without frame subsampling. Below we describe per-dataset evaluation protocols.

Failure penalty. For trajectory ATE tables, "–" denotes a missing or failed result. When computing averages, each missing entry is assigned the largest reported error in the same sequence column as a failure penalty.

KITTI. All 11 sequences (00–10) are evaluated with full frames.

vKITTI2. We evaluate all morning-condition scenes across the five virtual environments (Scene01, Scene02, Scene06, Scene18, Scene20). This is a training-overlap evaluation.

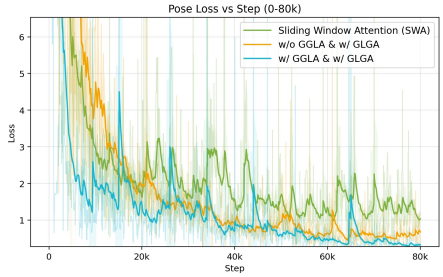


Figure 8: Training convergence under different attention mechanisms. In the legend, GGLA and GLGA denote Geometric Linear Attention (GLA) and Gated Window Attention (GWA), respectively; SWA denotes the sliding-window attention baseline. GLA with channel-wise gating converges faster and reaches a lower final loss.

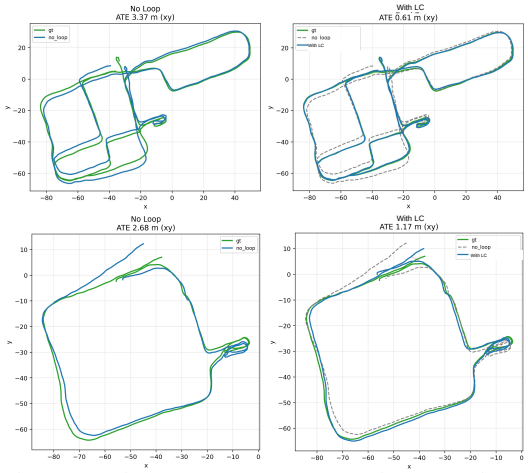


Figure 9: Effect of loop closure on long sequences. Loop closure reduces ATE on sequences with revisited regions while maintaining performance elsewhere.

7Scenes. We evaluate sequence 01 from each of the seven scenes. Among these evaluated sequences, those from Chess, Fire, Office, and RedKitchen are included in our training data, whereas those from Heads, Pumpkin, and Stairs are excluded.

Waymo Open. We select 9 segments not present in our training set: 163453191 (198 frames, 160 m), 183829460 (199 frames, 42 m), 315615587 (199 frames, 165 m), 346181117 (199 frames, 351 m), 371159869 (196 frames, 273 m), 405841035 (199 frames, 86 m), 460417311 (198 frames, 266 m), 520018670 (199 frames, 135 m), 610454533 (198 frames, 63 m). Although Waymo is part of our training data, these specific segments are held out to evaluate generalization on unseen driving scenes.

ScanNet++. We evaluate on 5 scenes: 419cbe7c11, 98b4ec142f, bb87c292ad, c24f94007b, ebc200e928. The evaluation segments from these five scenes are excluded from training.

Oxford Spires. We evaluate on 12 sequences. Since the ground-truth point clouds and images are in different coordinate systems, we perform image-to-ground-truth point cloud alignment. The number of aligned images varies across subsets, increasing evaluation difficulty. Per-sequence frame counts and trajectory lengths are shown in Tab. 8.

VBR. Following the LoGeR [61] setting, all 7 sequences are evaluated at full length (8,815 to 18,846 frames, up to 5.2 km).

TUM RGB-D and ETH3D. Standard evaluation protocols with full sequences.

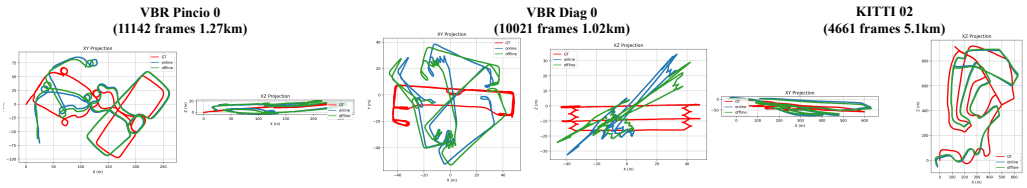


Figure 10: Failure cases on ultra-long sequences. Ground-truth trajectory (red), online prediction (blue), and loop-closure refined trajectory (green). Failures occur mainly in sequences with dense revisits or visually ambiguous regions.

E Additional Experimental Results

Tab. 8 reports per-sequence ATE on the 12 Oxford Spires evaluation sites. HorizonStream achieves the lowest average ATE among all online methods, with particularly large margins on long-trajectory sequences such as college5 and christ2.

Table 8: Quantitative comparison on Oxford Spires. We report ATE, where lower is better.

	Method	Oxford Spires ATE ↓												Avg.
		college2	college3	college4	college5	observ1	observ2	blenheim1	blenheim2	blenheim5	christ2	christ3	bodleian2	
		787 fr. 290 m	757 fr. 280 m	701 fr. 773 m	636 fr. 696 m	353 fr. 393 m	351 fr. 387 m	57 fr. 341 m	25 fr. 316 m	12 fr. 259 m	567 fr. 629 m	289 fr. 309 m	22 fr. 537 m	
Opt.-based	MASt3R-SLAM	15.97	31.89	–	–	20.05	21.44	–	45.48	50.62	–	–	–	37.73
	VGGT-SLAM	–	13.53	14.64	–	29.12	–	40.04	19.20	10.36	–	–	80.54	31.00
	COLMAP	0.06	0.05	0.05	0.32	0.17	0.26	0.21	0.06	–	39.55	14.99	–	15.57
	MASt3R-SfM	21.55	13.57	32.53	36.84	23.14	27.35	25.84	37.22	47.80	35.55	14.71	69.48	32.13
	DPVO	14.31	31.60	39.26	34.79	26.91	28.14	35.37	44.76	47.96	19.98	16.00	69.31	34.03
	DROID-SLAM	20.96	23.09	20.39	38.30	17.20	23.86	30.39	46.78	47.08	16.97	16.04	71.85	31.08
Offline	VGGT-Long	6.55	14.32	13.20	40.27	11.95	6.47	24.77	19.20	10.36	19.40	15.75	80.54	21.90
	FastVGGT	24.10	32.54	39.93	37.07	26.86	26.57	33.04	–	38.14	40.90	15.79	78.52	36.58
	LoGeR	6.76	9.37	5.46	7.66	6.16	5.62	26.82	26.90	32.92	19.33	3.91	73.46	18.70
	LoGeR*	2.79	11.92	3.97	9.05	4.78	4.71	27.34	20.61	21.80	8.67	2.86	71.04	15.79
Online Fwd.	CUT3R	30.68	23.73	31.31	30.80	25.32	26.21	31.15	37.09	38.71	37.25	14.33	62.68	32.44
	TTT3R	27.75	23.39	40.99	42.39	28.17	26.73	37.07	44.93	38.34	36.69	14.92	73.21	36.21
	STream3R	33.08	32.68	43.54	41.82	28.40	28.37	36.21	41.72	31.96	44.98	15.47	72.60	37.57
	StreamVGGT	31.97	31.09	43.55	42.15	29.09	28.24	36.08	42.57	30.16	41.23	15.74	75.18	37.25
	InfiniteVGGT	25.71	27.24	27.94	28.33	25.81	24.04	35.69	44.49	19.33	38.59	12.83	71.81	31.82
	Lingbot-Map	2.17	3.61	19.99	12.01	9.99	6.23	8.23	14.59	39.67	15.79	12.86	40.38	15.46
	Ours	2.81	0.76	10.87	2.84	1.43	2.17	6.71	11.62	33.53	4.70	3.53	31.59	9.38
	Ours w/ LC	1.05	0.70	9.85	0.88	1.08	1.16	6.71	11.62	33.53	4.70	1.68	31.59	8.71

Loop closure. Fig. 9 shows the effect of the optional loop-closure module on long sequences. Loop closure reduces ATE on sequences with revisited regions while maintaining comparable performance.

Memory and runtime scaling. Fig. 7 reports peak GPU memory and wall-clock time as sequence length grows from 200 to 10,000 frames. HorizonStream maintains nearly constant peak memory and scales smoothly, while competing methods either run out of memory or exhibit super-linear runtime growth.

Training convergence. Fig. 8 compares training loss curves when using different attention mechanisms for cross-window propagation. Geometric Linear Attention with channel-wise gating converges faster and reaches a lower final loss than the ungated and softmax-attention variants, reflecting the benefit of bounded multi-timescale retention during training.

Failure cases. Fig. 10 shows representative failure cases on ultra-long sequences. Errors mainly occur in sequences with dense revisits or visually ambiguous regions, where the fixed-size recurrent state does not preserve sufficient fine-grained information for precise relocalization. The optional loop-closure module partially mitigates these failures.

E.1 Channel-to-Geometry Linear Probing

Linear probing of recurrent geometric states. We further examine whether frozen Geometric Linear Attention states contain linearly decodable geometric information. For each chunk, we extract 1024-dimensional state features from all four Geometric Linear Attention layers. We train ridge regressors on KITTI sequences 00 and 02, and evaluate on the held-out sequence 05. The probes predict four geometric error targets: local translation error, long-range scale log error, long-range translation error, and segment scale log error.

Fig. 11 shows that segment-scale log error is the most reliably predictable target from frozen Geometric Linear Attention states, suggesting that the recurrent state contains measurable metric-related information.

Fig. 12 analyzes where the segment-scale signal resides across retention bands. The probe weights are distributed across short-, medium-, and long-retention channels, rather than being concentrated in a single band. This supports the view that metric evidence is represented across the learned retention spectrum and benefits from multi-timescale propagation.

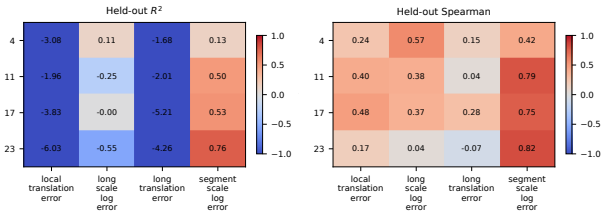


Figure 11: Linear probing of frozen Geometric Linear Attention states. Ridge regressors trained on chunk-level state features predict four geometric error targets. Segment-scale log error is the most reliably predictable, suggesting that the recurrent state encodes measurable metric information. Other targets are less consistently captured by a linear probe.

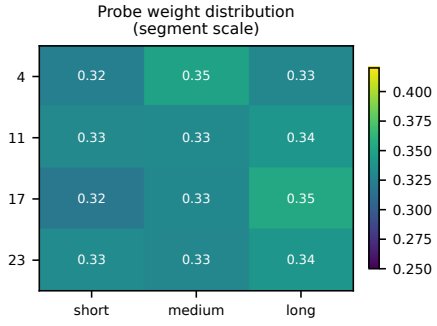


Figure 12: Probe weight attribution by retention band (segment scale target). Weights are nearly uniform across short / medium / long, showing that scale signal is distributed rather than band-specific.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes].

   Justification: The abstract and introduction state the paper's main claims, including the geometric evidence influence kernel formulation, the Horizon-Stream architecture, and its long-sequence reconstruction results. These claims are supported by the method, experiments, ablations, and limitations discussed in the paper.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes].

   Justification: The paper includes a limitations discussion covering dynamic foreground objects, extremely long sequences with repeated revisits, and the parameterization of the optional loop-closure module.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

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

Answer: [Yes].

Justification: The paper provides formal analysis and proofs in the appendix for the attention dilution behavior, open-form operator failure, and bounded influence of the recurrent geometric state.

Guidelines:

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

4. **Experimental result reproducibility**

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

Answer: [Yes].

Justification: The paper describes the model architecture, training setup, datasets, evaluation metrics, baselines, and implementation details needed to reproduce the main experimental results. Additional details are provided in the appendix.

Guidelines:

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

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

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

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

   Answer: [Yes].

   Justification: The code, evaluation scripts, and reproduction instructions have been publicly released.

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes].

   Justification: The paper reports the training and testing protocols, including model configuration, datasets, evaluation metrics, and ablation settings. Additional implementation and hyperparameter details are included in the appendix.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [No].

   Justification: The paper reports deterministic benchmark results on standard 3D reconstruction datasets and does not include error bars over multiple training runs due to computational cost. The main conclusions are supported by consistent performance across multiple datasets, sequences, and ablation studies.

Guidelines:

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

8. **Experiments compute resources**

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

   Answer: [Yes].

   Justification: The paper reports the computational setup used for training and evaluation, including GPU resources and memory/runtime characteristics. The method's linear-time and constant-memory behavior is also discussed.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes].

   Justification: The work uses public datasets and focuses on 3D reconstruction methodology. The authors have reviewed the NeurIPS Code of Ethics and identify no deviation from it.

   Guidelines:

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

10. **Broader impacts**

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

Answer: [Yes].

Justification: The paper discusses potential positive impacts for robotics, autonomous driving, and embodied perception, as well as possible risks related to privacy-sensitive 3D reconstruction and surveillance-like misuse.

Guidelines:

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

11. **Safeguards**

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

Answer: [N/A].

Justification: The paper does not release high-risk generative models, scraped datasets, or models intended for direct sensitive-domain deployment.

Guidelines:

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

12. **Licenses for existing assets**

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

Answer: [N/A].

Justification: The paper does not introduce a new dataset or benchmark. Any released code or model checkpoints will be accompanied by documentation.

Guidelines:

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

13. **New assets**

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

Answer: [N/A].

Justification: The paper does not involve crowdsourcing experiments or research with human subjects.

Guidelines:

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

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

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

Answer: [N/A].

Justification: The paper does not involve crowdsourcing experiments or research with human subjects.

Guidelines:

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

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

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

Answer: [N/A].

Justification: The paper does not involve human subjects, participant studies, or crowd-sourced data collection.

Guidelines:

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

16. **Declaration of LLM usage**

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

Answer: [N/A].

Justification: LLMs are not used as an important, original, or non-standard component of the core method.

Guidelines:

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