MAMBA-VGGT: EXPLICIT SPATIAL STATE FOR CROSS-WINDOW VIDEO GEOMETRY

Anonymous authors
Paper under double-blind review

ABSTRACT

Recovering camera motion and scene geometry from long videos requires strong multi-view reasoning and effective reuse of past observations. Pretrained geometric transformers provide powerful multi-view priors, but their global attention becomes increasingly costly as the input grows. Windowed inference limits this cost while interrupting feature-level communication across windows, which output-level geometric stitching cannot restore. We introduce Mamba-VGGT, a recurrent adapter that extends a frozen Visual Geometry Grounded Transformer with fixed-capacity cross-window memory. A Mamba-based two-dimensional selective scan (SS2D) jointly processes the incoming state and current patch tokens, producing coupled residual feature updates and memory writes. This design retains pretrained within-window attention while allowing subsequent windows to use historical context. State decay and periodic segment resets control state accumulation during long-sequence inference. On KITTI-360 sequences with 2,000 sampled frames, Mamba-VGGT reduces mean absolute trajectory error by 42.3% and mean LiDAR-supported reconstruction distance by 56.3% relative to LoGeR with sliding-window attention under the evaluated inference configuration. It also achieves lower trajectory error on VBR inputs of 8,000 and 10,000 frames under matched window and reset schedules. Mixer-replacement experiments favor SS2D over a parameter-matched pointwise MLP under the tested training setup, while state-propagation ablations show that carried context further improves trajectory estimation. Together, these results support joint spatial adaptation and cross-window memory as an effective approach to extending frozen geometric models for long-sequence camera estimation and reconstruction.

1 INTRODUCTION

Recovering camera motion and scene geometry from image sequences is central to visual perception. Feed-forward geometric models such as DUSt3R, MASt3R, Fast3R, and VGGT encode strong multi-view priors in pretrained networks (Wang et al., 2024; Leroy et al., 2024; Yang et al., 2025; Wang et al., 2025a). VGGT alternates frame-wise and global attention to jointly infer cameras, depth, and point maps. Extending this reasoning to long videos, however, becomes costly when global attention spans the entire input. Short overlapping windows bound the per-forward context while retaining bidirectional reasoning, but independent processing discards intermediate features between windows. Geometric stitching aligns the resulting predictions without restoring feature-level communication across windows.

Recent recurrent and hybrid-memory approaches demonstrate the value of carrying historical context through persistent states, cached features, or fast-weight memory (Wang et al., 2025b; Zhuo et al., 2025; Zhang et al., 2026). Building on these advances, we focus on extending a strong pretrained geometric model through compact recurrent adaptation while keeping its backbone and prediction heads frozen. This requires a memory mechanism that can both condition current-window features on earlier observations and incorporate new evidence for subsequent predictions.

We introduce *Mamba-VGGT*, which connects bidirectional VGGT windows through fixed-capacity patch-grid states at four network stages. At each stage, the incoming state and current patch tokens are concatenated and processed by SS2D, a four-directional spatial selective scan based on Mamba and VMamba (Gu & Dao, 2024; Liu et al., 2024). Two projections of the resulting joint representation

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

produce a token residual and a state increment. This shared, content-dependent transformation couples the use of historical context for current geometry estimation with the writing of new context for later windows. The patch-grid state serves as a structured latent feature memory whose capacity remains independent of sequence length.

Spatial adaptation and temporal persistence play distinct roles in this design: SS2D mixes features within each frame grid, while the external state propagates context across window boundaries. Zero-initialized residual projections preserve the original VGGT function at initialization, and subsequent training optimizes only the added modules, amounting to 0.63% of the pretrained model's parameter count. During long-sequence inference, state decay and periodic segment resets control state accumulation. Together, these components provide a compact adaptation path while retaining VGGT's pretrained attention blocks and task heads.

We evaluate camera estimation and reconstruction on KITTI-360 inputs of 300, 1000, and 2000 frames, and further assess long-sequence camera estimation on VBR inputs of 8000, 10,000 and more frames. On the 2000-frame KITTI-360 cases, Mamba-VGGT reduces mean absolute trajectory error and reconstruction Chamfer distance by 35.7% and 43.8%, respectively, relative to matched windowed VGGT. On the evaluated 10,000-frame VBR sequence, it reduces absolute trajectory error by 19.5% relative to LoGeR under matched window and reset settings. Mixer-replacement experiments favor SS2D over a parameter-matched pointwise MLP under the tested training setup, while state-propagation ablations show that carried context further improves trajectory estimation. Our contributions are:

- We introduce Mamba-VGGT, a compact recurrent adaptation architecture that connects bidirectional VGGT windows through fixed-capacity feature states while keeping the pretrained backbone and prediction heads frozen.
- We develop a joint state–token SS2D mechanism that couples residual feature adaptation with cross-window state writing, complemented by state decay and segment resets for long-sequence inference.
- We evaluate long-sequence geometry on KITTI-360 and VBR, and analyze the contributions of historical context and spatial adaptation through state-propagation ablations and mixer-replacement experiments.

2 Related Work

Feed-forward geometric models. DUSt3R predicts point maps from image pairs, and MASt3R extends this formulation with stronger matching representations (Wang et al., 2024; Leroy et al., 2024). Fast3R and VGGT broaden feed-forward reconstruction to multi-view inputs, with VGGT alternating frame-wise and global attention to predict cameras and scene geometry (Yang et al., 2025; Wang et al., 2025a). These models motivate retaining pretrained multi-view reasoning when extending inference to longer inputs. We use VGGT as a frozen within-window backbone and add a trainable feature-adaptation path, rather than replacing its attention blocks or prediction heads.

Persistent context for online reconstruction. Spann3R maintains spatially addressable features for incremental reconstruction, whereas CUT3R propagates a learned latent state for continuous camera and geometry prediction (Wang & Agapito, 2025; Wang et al., 2025b). StreamVGGT adapts bidirectional VGGT to causal inference and reuses historical keys and values (Zhuo et al., 2025). These approaches illustrate different ways to expose past observations to current predictions. Our design instead operates between bidirectional windows: a fixed-size activation state conditions patch features through an added recurrent module. It does not retain a sequence-length-dependent token history or assign persistent world coordinates to state slots.

Test-time training and hybrid memory. Test-time training (TTT) sequence models use fast weights as an adaptive hidden state (Sun et al., 2025). In geometric reconstruction, TTT3R interprets recurrent inference as online learning and adapts updates using alignment confidence (Chen et al., 2025). ZipMap uses fast-weight memory for linear-time stateful reconstruction, while LoGeR combines TTT memory with sliding-window attention (Jin et al., 2026; Zhang et al., 2026). LoGeR is closely related in combining local reasoning with persistent context. We take the compact read–

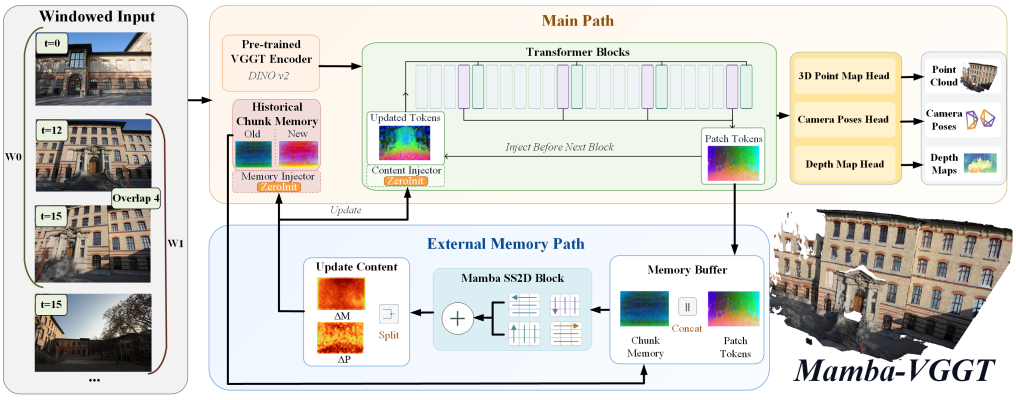


Figure 1: Mamba-VGGT. Bidirectional VGGT windows share four patch-grid feature states. A joint SS2D transform produces residual feature injection and the state increment carried to the next window. The backbone and original prediction heads remain frozen.

update–pass interface as an inspiration and instantiate it with patch-grid activations and an SS2D adapter around frozen VGGT.

3 Method

3.1 Architecture

Given an ordered video $\mathcal{I} = \{I_t\}_{t=1}^{T}$, we divide it into overlapping windows $\mathcal{W}_m$ of $S = 16$ frames with an overlap of $O = 4$. Each window is processed by the pretrained bidirectional VGGT, augmented with four recurrent adaptation modules at zero-based block indices $\mathcal{L}_{\text{mem}} = \{3, 10, 16, 22\}$. Each module operates after the corresponding frame-wise and global attention operations and maintains its own patch-grid state across windows.

At each stage, a joint SS2D mixer combines the incoming state with current-window patch features to produce a token residual and a state increment. The residual conditions subsequent VGGT layers on historical context, while the state is updated at the window boundary and passed to the next window. All frames within a window read the same incoming state. The pretrained backbone and prediction heads remain frozen, and only the added adaptation modules are trained. Figures 1 and 2 illustrate the overall architecture and the joint state–token mixer, respectively.

3.2 Joint State–Token Adaptation

Let $X_{m,s}^{\ell} \in \mathbb{R}^{B\times H_p\times W_p\times C}$ denote the patch tokens of frame s at tap ℓ before residual injection, and $M_{m-1}^{\ell} \in \mathbb{R}^{B\times H_p\times W_p\times K}$ the incoming state. Broadcasting the same state to each frame, we concatenate it with current features and split the joint SS2D output into state and token channels:

$$J_{m,s}^{\ell} = [M_{m-1}^{\ell} \| X_{m,s}^{\ell}], \tag{1}$$

$$[Y_{m,s}^{M,\ell}, Y_{m,s}^{X,\ell}] = \text{Split}_{K,C}\big(\Phi_{\ell}(J_{m,s}^{\ell})\big)\,. \tag{2}$$

The state is a geometric *feature* prior, not an explicit 3D map: grid slots are not assigned fixed world points. Accordingly, no output pose/scale transform is applied to the latent state. Subsequent frozen attention and prediction heads recover geometry from state-conditioned patch tokens; camera and register tokens bypass the SS2D path.

Selective spatial mixing. The mixer Φ_{ℓ} follows Mamba and VMamba (Gu & Dao, 2024; Liu et al., 2024). A gated projection and depthwise convolution feed four traversals: row-major, column-major, and their reverses. At spatial position p in direction d, the input determines selective coefficients $(\Delta_{d,p}, B_{d,p}, C_{d,p})$. The scan uses

$$h_{d,p} = \exp(\Delta_{d,p}A_d) \odot h_{d,p-1} + \Delta_{d,p}B_{d,p} \odot u_{d,p}, \qquad A_d = -\exp(A_d^{\log}), \tag{3}$$

$$y_{d,p} = C_{d,p}h_{d,p} + D_d \odot u_{d,p}. \tag{4}$$

Here $B_{d,p}$ and $C_{d,p}$ are SSM coefficients, not the batch dimension or cross-window state. The directional outputs are restored to a common grid, summed, normalized, gated, and projected back to

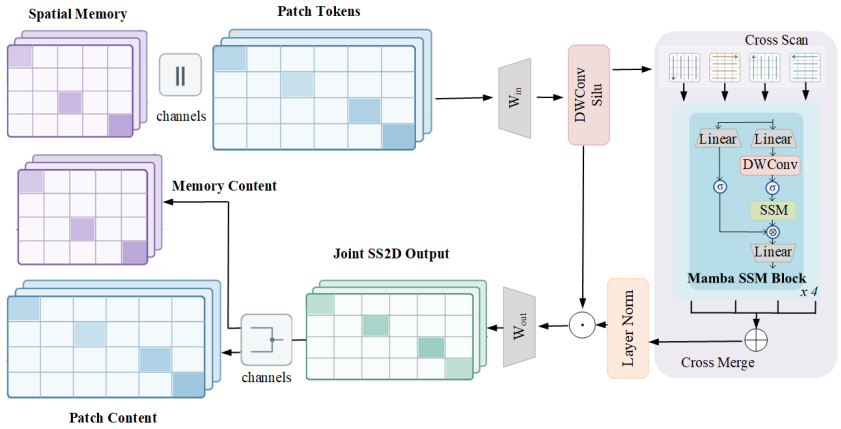


Figure 2: Joint state–token adaptation. Incoming spatial memory and current patch features are mixed by four-direction SS2D before being split into memory and patch content. Separate zero-initialized output projections produce token residuals and state increments.

$C + K$ channels. We use $C = 1024$, $K = 64$, internal width 256, and scan-state dimension 16. The same mixer operates independently on each frame grid: the scan state h is transient within that call, whereas only the external state M persists between windows. Expanded projection equations are in Appendix D.1.

Residual injection and one boundary update. The token branch modifies the current features through

$$\widetilde{X}^{\ell}_{m,s} = X^{\ell}_{m,s} + W^{\ell}_{X} Y^{X,\ell}_{m,s}. \tag{5}$$

Both the token residual and the write content come from the same incoming state and pre-injection features. Let $w_{m,s} = 1$ for a frame committed at its first appearance and 0 for repeated overlap frames. We pool only new frames and update the state once at the window boundary:

$$\overline{Y}^{M,\ell}_{m} = \frac{\sum_s w_{m,s} Y^{M,\ell}_{m,s}}{\max(1, \sum_s w_{m,s})}, \tag{6}$$

$$\Delta M^{\ell}_{m} = W^{\ell}_{M} \overline{Y}^{M,\ell}_{m}, \qquad M^{\ell}_{m} = \alpha^{\ell}_{m} M^{\ell}_{m-1} + \Delta M^{\ell}_{m}. \tag{7}$$

All frames in one window read the unchanged M^{ℓ}_{m-1}; the new state is exposed only to the next window. With no newly committed frame, the state is copied exactly, bypassing writer bias and decay. The base update uses $\alpha = 1$; the inference-only decay policy is specified below. Zero-initializing W^{ℓ}_{X} and W^{ℓ}_{M}, including their biases, gives

$$\widetilde{X}^{\ell}_{m,s} = X^{\ell}_{m,s}, \qquad M^{\ell}_{m} = 0 \quad \text{at initialization}, \tag{8}$$

so the untrained adapter reproduces windowed VGGT. After training, *adapter off* bypasses all added modules, whereas *zero inter-window state* retains their transformations but discards the propagated state. These controls need not produce the same predictions.

3.3 Windowed Inference

Inference starts from $M^{\ell}_{0} = 0$. For every window, Mamba-VGGT reads the fixed incoming states, jointly predicts cameras and geometry with the original heads, then commits the states for its successor. The stride $S - O$ preserves temporal order and bidirectional reasoning within each window. Predictions have local gauges, so overlap-based registration maps them into a common output frame:

$$\widehat{P}^{\text{global}}_{t} = s_m R_m \widehat{P}_{m,t} + \mathbf{t}_m. \tag{9}$$

The transform is estimated from predicted overlap frames; KITTI uses fixed-scale SE(3), $s_m = 1$, shared with VGGT-W16. When two windows predict the same frame, the later prediction is retained. Stitching aligns outputs; it does not transform the latent state or use GT.

State decay and segment resets. For medium/long inference, state decay (leak5) attenuates old state at the late taps after five windows. Writing q_m for the window index within the current segment,

$$\alpha_m^\ell = \begin{cases} 0.8, & \ell \in \{16, 22\},\ q_m > 5 \text{ and new frames are committed,} \\ 1, & \text{otherwise.} \end{cases} \tag{10}$$

This is exponential forgetting, not a five-window FIFO; a row without new frames retains its state. For KITTI 1000/2000-frame runs, segment resets (reset300) additionally start a new segment every 300 new observations, reinitializing all four states and the leak counter. Segments retain four boundary-overlap frames and are registered with predicted-overlap SE(3). The 300-frame runs have no additional reset. Leak and reset are runtime interventions, not additional training objectives; per-dataset settings are in Appendix A. Appendices D.4 and D.5 provide leak5 state-amplitude and local-geometry diagnostics, and a complete-trajectory reset300 comparison, respectively.

For $Q = H_p W_p$, four propagated states occupy $\mathcal{O}(|\mathcal{L}_{\text{mem}}|BQK)$ elements, independent of video length. With internal width D and scan-state dimension N, the additional scan cost per window is $\mathcal{O}(|\mathcal{L}_{\text{mem}}|BS_mQDN)$. Fixed-window VGGT attention keeps its original cost. These bounds exclude total runtime memory, stored outputs, and training activations.

4 Experiments

4.1 Setup and Baselines

Training and evaluation. Mamba-VGGT trains 7.915M added parameters (0.63% of VGGT), with a 32-frame adaptation stage followed by 64-frame continuation. Training mixes CO3D, ARKitScenes, WildRGB-D, BlendedMVS, DL3DV-10K, and ScanNet++ (Reizenstein et al., 2021; Dehghan et al., 2021; Xia et al., 2024; Yao et al., 2020; Ling et al., 2024; Yeshwanth et al., 2023). The objective combines camera, depth, and point-map supervision with cross-window overlap consistency; loss weights and optimization details are in Appendix A.1. The main text reports 300-frame KITTI-360 camera tracking and reconstruction, averaged equally over four input blocks. Since KITTI-360 provides sparse LiDAR depth, the depth comparison instead uses NRGBD. Sampling rules and exact input lengths are given in Appendix A.2. At 1000 and 2000 frames, we uniformly sample the first 2500 valid observations of test_2 and test_3. Both lengths cover the same physical span at different densities, not a doubled route. Representative 32-frame results appear in Section 4.2; complete short-context tables and visualizations remain in Appendix B.

Baseline configurations. VGGT-W16 disables adapters and uses matched 16/4 windows, segment boundaries, and KITTI SE(3) stitching. ZipMap uses native 12-frame TTT updates, matching the 12-new-frame stride of the 16/4 window schedule. LoGeR is evaluated with matched 16/4 windows, both TTT and SWA, and its native full-image resize (255,000-pixel budget). For long KITTI runs, LoGeR resets TTT every 300 new observations while preserving SWA history and uses native rigid segment registration. CUT3R, Spann3R, and StreamVGGT retain native recurrent/causal inference; TTT3R uses reset100. All methods share sampled frame identities but retain their required preprocessing: KITTI inputs are 518×140 for Mamba-VGGT/VGGT and 980×252 for LoGeR. This is a comparison of stated inference configurations, not equal-resolution or equal-compute execution. At 2000 frames StreamVGGT exhausts H200 memory on both cases; we report OOM rather than shorten or reset its sequence. Detailed execution protocols are recorded in Appendix E.

Metrics and support. ATE is translation RMSE after one whole-sequence Sim(3); RPE_r and RPE_t are adjacent-frame rotation and translation-direction errors in degrees. Depth uses one median scale per sequence, reporting AbsRel and δ_1. Reconstruction backprojects predicted depth with predicted intrinsics and poses. KITTI Acc., Comp., and their mean CD measure nearest-neighbor distances on common positive, finite LiDAR-supported predictions, with disjoint 20%/80% subsets for fitting a single Sim(3) and scoring on a 0.1 m voxel grid. These are conditional observed-support distances, not dense scene coverage. Long evaluations preselect every fourth projected LiDAR ray, then intersect successful methods; OOM rows are excluded from that intersection, not assigned zero error. No ICP or per-window GT correction is used.

Table 1: 32-frame pose (ATE), depth (AbsRel), and reconstruction (CD), all ↓. ATE uses meters on ETH3D and native dataset units on RE10K; CD uses meters. Results and protocols follow Tables B1–B3: ETH3D (Schöps et al., 2017), KITTI-360 (Liao et al., 2023), and RE10K (Zhou et al., 2018). CD averages Acc./Comp.; bold marks column bests.

	ATE↓		AbsRel↓		CD (m)↓	
Method	ETH3D	RE10K	ETH3D	KITTI-360	ETH3D	KITTI-360
VGGT-W16 (Wang et al., 2025a)	1.730	0.02120	0.04480	0.09460	0.7619	0.4931
Spann3R (Wang & Agapito, 2025)	2.474	0.08960	0.1489	0.2112	13.09	0.6656
CUT3R (Wang et al., 2025b)	2.407	0.04740	0.08340	0.1798	1.185	1.030
TTT3R (Chen et al., 2025)	1.859	0.04930	0.07130	0.1666	1.207	0.7754
StreamVGGT (Zhuo et al., 2025)	3.211	0.1018	0.1245	0.1479	1.177	0.6535
ZipMap-W12 (Jin et al., 2026)	2.538	0.04280	0.04680	0.08690	1.175	0.4607
Mamba-VGGT	**0.5457**	**0.01670**	**0.03460**	**0.08420**	**0.6650**	**0.4322**

4.2 Representative 32-Frame Geometry

Mamba-VGGT achieves the lowest errors in all six columns of Table 1. Relative to VGGT-W16, ETH3D AbsRel/ATE/CD decrease by 22.8%/68.5%/12.7%, and KITTI CD by 12.4%. Figure 3 shows qualitative examples; complete results are in Appendix B.

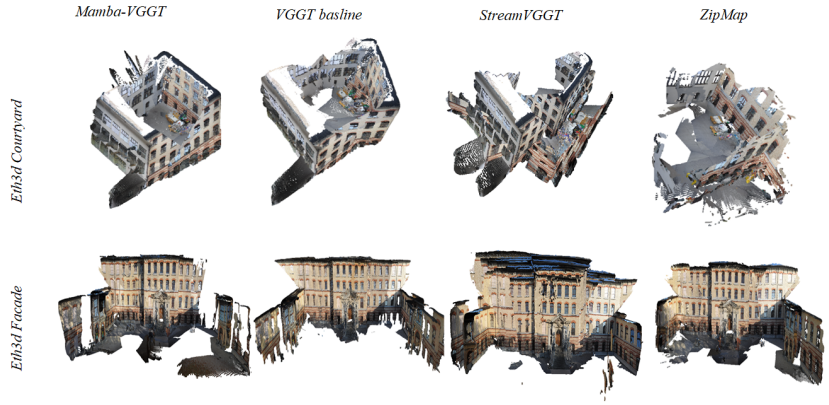


Figure 3: 32-frame ETH3D *courtyard* (top) and *facade* (bottom), showing visible structure and layout. Columns: Mamba-VGGT, VGGT baseline, StreamVGGT, ZipMap. These are qualitative comparisons, not additional quantitative metrics; further scenes appear in Figure B1.

4.3 Medium-Length Geometry

Table 2: Medium-length geometry. KITTI-360 pose and reconstruction are equal-weight means over four blocks with 300 sampled frames each. The Mamba-VGGT row is the selected main checkpoint with normal state propagation. The depth columns use NRGBD, not KITTI LiDAR. Distances are meters; RPE values are degrees; δ_1 is a fraction. Table values use four significant figures.

	KITTI-360 pose (Liao et al., 2023)			NRGBD depth (Azinovic et al., 2022)		KITTI-360 reconstruction (Liao et al., 2023)		
Method	ATE↓	RPE_r ↓	RPE_t ↓	AbsRel↓	δ_1 ↑	Acc.↓	Comp.↓	CD↓
VGGT-W16 (Wang et al., 2025a)	27.56	0.4677	7.209	0.03700	**0.9977**	5.920	9.693	7.807
StreamVGGT (Zhuo et al., 2025)	108.0	4.987	76.27	0.04560	0.9975	11.23	33.22	22.22
ZipMap-W12 (Jin et al., 2026)	68.85	1.853	72.44	0.03320	0.9952	11.57	13.21	12.39
TTT3R (Chen et al., 2025)	71.19	2.909	45.70	0.09200	0.9739	23.24	32.43	27.83
CUT3R (Wang et al., 2025b)	97.43	5.328	59.50	0.09930	0.9733	18.70	36.10	27.40
Spann3R (Wang & Agapito, 2025)	110.6	11.95	76.28	0.03870	0.9919	14.86	25.23	20.05
LoGeR + SWA (Zhang et al., 2026)	49.76	2.109	24.87	**0.02150**	0.9975	10.32	9.868	10.09
Mamba-VGGT	**21.22**	**0.3892**	**6.639**	0.03100	0.9969	**5.421**	**5.892**	**5.657**

Mamba-VGGT gives the lowest mean ATE and both RPE measures among the displayed methods at 300 frames (Table 2). Mean ATE is 21.2157 m, compared with 27.5629 m for VGGT-W16 and

49.7615 m for LoGeR + SWA, reductions of 23.0% and 57.4%. Mean reconstruction CD is 5.6568 m, 27.5% and 44.0% lower, respectively. On NRGBD, Mamba-VGGT has AbsRel 0.0310 versus 0.0370 for VGGT-W16, a 16.2% reduction, and ranks second behind LoGeR (0.0215). Threshold accuracy is lower than VGGT-W16 ($\delta_1 = 0.9969$ versus 0.9977). Figure 4 shows three representative blocks.

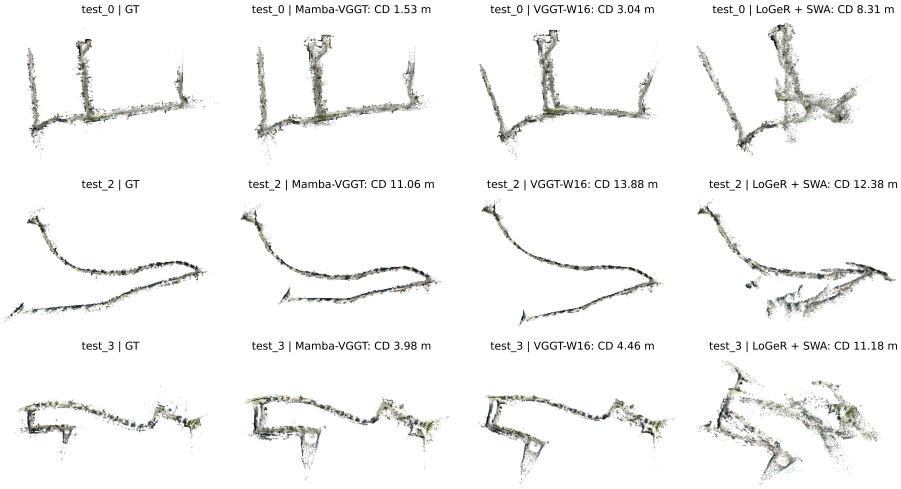


Figure 4: KITTI-360, 300 sampled frames per block. Rows: test_0, test_2, test_3; columns: GT, Mamba-VGGT, VGGT-W16, LoGeR + SWA. Each row shares RGB colors, held-out LiDAR observations, view, and scale. No axes, clipping, or independent rescaling; all four blocks appear in Figure C1. Labels give per-block CD (m), not the four-block means in Table 2.

4.4 Long-Sequence Pose and Reconstruction

Table 3: Long-sequence camera tracking on KITTI-360 (Liao et al., 2023). ATE (m)↓ after one global Sim(3) per sequence. Both input lengths sample the same first 2500 valid observations. Mamba-VGGT/VGGT and LoGeR use their stated reset300 schedules; TTT3R uses reset100. Additional execution details are in Appendix E.

	1000 frames		2000 frames	
Method	test_2	test_3	test_2	test_3
VGGT-W16 (Wang et al., 2025a)	51.34	48.67	48.57	53.01
StreamVGGT (Zhuo et al., 2025)	166.1	130.9	OOM	OOM
ZipMap-W12 (Jin et al., 2026)	154.8	105.4	158.6	138.7
TTT3R (Chen et al., 2025)	124.3	124.2	128.3	109.6
CUT3R (Wang et al., 2025b)	170.3	137.4	184.7	125.5
Spann3R (Wang & Agapito, 2025)	197.2	139.2	197.4	135.3
LoGeR + SWA (Zhang et al., 2026)	81.16	58.15	68.62	44.64
Mamba-VGGT	**41.15**	**22.52**	**46.65**	**18.68**

Tables 3–4 extend the evaluation to 1000 and 2000 sampled frames. Mamba-VGGT has the lowest ATE and all three reconstruction distances among successful methods for each case and length. Across the two 1000-frame cases, mean ATE is 31.8366 m, 36.3% below VGGT-W16 and 54.3% below LoGeR; mean CD is 8.3725 m, lower by 44.7% and 37.8%. At 2000 frames, mean ATE is 32.6660 m and CD is 8.9268 m: ATE improves by 35.7%/42.3% and CD by 43.8%/56.3% relative to VGGT-W16/LoGeR. Figure 5 visualizes the same held-out observations for test_2 at 2000 frames.

Beyond KITTI: VBR. Mamba-VGGT achieves the lowest ATE in all four W16/O4 settings (Table 5). On Ciampino, ATE decreases relative to LoGeR by 4.7% with 12,000 sampled frames and 13.7% with all 18,846 frames. The advantage thus persists without temporal subsampling. Protocols are detailed in Appendix E.1; Figure 6 shows qualitative Campus results.

With LoGeR's reported VBR schedule (W48/O3, TTT resets every five windows), its Campus/DIAG ATEs are 64.95/33.62 m, versus 61.21/32.86 m for Mamba-VGGT under W16/O4. Thus, the advantage persists with LoGeR's reported schedule, rather than only under the W16/O4 configuration.

Table 4: Long-sequence KITTI-360 (Liao et al., 2023) reconstruction. Acc., Comp., and CD (m)↓ use shared held-out LiDAR rays and a single fitted Sim(3) per sequence. Support is intersected across eight successful methods at 1000 frames and seven at 2000 frames; StreamVGGT fails with OOM at 2000 frames. Values are conditional observed-support distances, not dense scene coverage.

	1000 frames						2000 frames					
	test_2			test_3			test_2			test_3		
Method	Acc.	Comp.	CD	Acc.	Comp.	CD	Acc.	Comp.	CD	Acc.	Comp.	CD
VGGT-W16 (Wang et al., 2025a)	16.48	18.15	17.31	11.10	14.85	12.97	14.72	20.53	17.62	11.07	17.18	14.13
StreamVGGT (Zhuo et al., 2025)	16.03	38.50	27.27	9.875	31.19	20.53	OOM	OOM	OOM	OOM	OOM	OOM
ZipMap-W12 (Jin et al., 2026)	23.97	44.23	34.10	13.61	22.77	18.19	20.97	35.13	28.05	12.78	76.45	44.62
TTT3R (Chen et al., 2025)	18.09	65.83	41.96	15.66	54.60	35.13	16.69	63.61	40.15	17.87	40.35	29.11
CUT3R (Wang et al., 2025b)	22.54	40.60	31.57	23.05	81.05	52.05	20.04	76.28	48.16	14.00	35.20	24.60
Spann3R (Wang & Agapito, 2025)	18.94	39.21	29.08	12.87	45.36	29.12	28.20	81.93	55.07	11.94	33.72	22.83
LoGeR + SWA (Zhang et al., 2026)	15.13	15.16	15.14	12.07	11.48	11.77	29.25	29.79	29.52	12.06	10.62	11.34
Mamba-VGGT	**10.40**	**8.827**	**9.613**	**7.131**	**7.134**	**7.132**	**11.76**	**11.15**	**11.45**	**6.330**	**6.472**	**6.401**

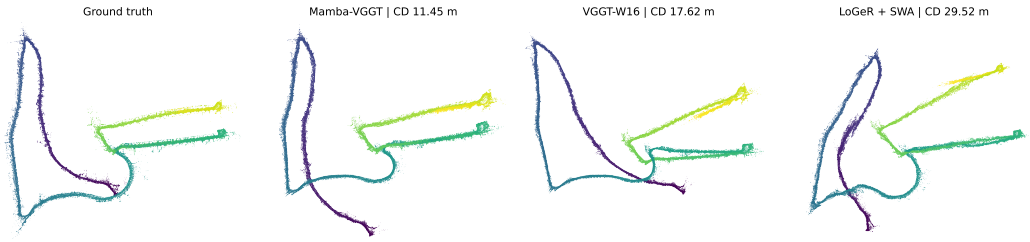


Figure 5: KITTI-360 test_2, 2000 frames. GT and predictions share held-out LiDAR observations, frame-index colors, and a common view and scale; axes are omitted. Each prediction uses only its evaluation Sim(3), with no clipping or smoothing. Labels show CD (m). These are observed-support clouds, not dense coverage; all successful baselines appear in Appendix C.

Table 5: VBR tracking (Brizi et al., 2024). ATE (m)↓ after one whole-sequence Sim(3). All methods use W16/O4 and reset300; Mamba-VGGT uses leak5. The final two columns compare sampled and full-sequence Ciampino inputs. Native resizing is retained (Appendix E.1).

Method Frames	Campus 10,000	DIAG 8,000	Ciampino 12,000 (sampled)	Ciampino 18,846 (full)
VGGT-W16 (Wang et al., 2025a)	85.39	33.18	172.6	167.1
LoGeR + SWA (Zhang et al., 2026)	76.04	34.29	171.0	180.9
Mamba-VGGT	**61.21**	**32.86**	**162.9**	**156.2**

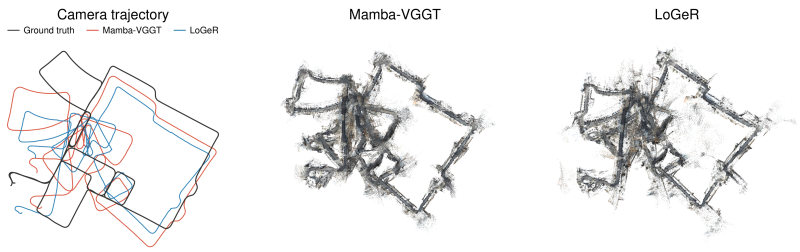


Figure 6: VBR Campus, 10,000 RGB-adaptive sampled frames. Left: camera trajectory. Middle/right: Mamba-VGGT/LoGeR point clouds from the same 10,000-frame input. Each prediction receives one pose Sim(3) fitted over its evaluated input sequence, also applied to its cloud. Cloud panels share image-grid samples, view, and scale, without ICP or error clipping; point subsampling is for display only.

VBR end-to-end resources. On Campus (10,000 frames, W16/O4), Mamba-VGGT takes 650.02 s end to end versus 1086.76 s for LoGeR (40.2% less), with 41.7%/33.9% lower process-level PyTorch peak allocated/reserved memory. All methods offload outputs to CPU. Native resolutions differ, so these measurements compare implementations, not state representations alone. They do not establish constant total storage. Full measurements and timing scope are in Appendix E.1.

4.5 Mixer and State-Propagation Ablations

Initialization and controls. Table 6 separates a matched mixer comparison from state-propagation interventions. Three continuation variants inherit compatible backbone, head, token-residual, and state-write parameters from the same 32f checkpoint, but initialize their mixers afresh. The matched recurrent SS2D and pointwise $1088 \to 424 \to 1088$ MLP (SiLU and LayerNorm) use the same recurrent training, writer updates, data, optimization, and validation checkpoint-selection rule. Their mixer parameter counts differ by 0.082%. A separate SS2D zero-history continuation freezes its inherited writer and fixes window input/output states to zero. The selected main checkpoint retains its trained mixer and is reported separately (Appendix D.2).

Matched mixer comparison. With state propagation enabled at inference, matched recurrent SS2D achieves ATEs of 36.16/18.24 m on test_2/test_3, versus 47.6816/45.7828 m for the pointwise MLP: reductions of 24.2%/60.2%. Controlling mixer initialization history and the continuation recipe, this comparison favors SS2D over the similarly sized pointwise MLP. It evaluates the complete mixer and excludes the zero-history and selected main checkpoints.

Table 6: KITTI-360 (Liao et al., 2023) ablations: ATE (m)↓ on identical 1000-frame inputs. Top: freshly initialized mixers with matched recurrent training, writer updates, data, optimization, and validation selection. Middle/bottom: zero-history continuation/selected main checkpoint; each on/zero pair shares inference weights. Bold marks block bests.

Branch / mixer	Mixer params. per tap	Training state	Inference state	test_2	test_3
Pointwise MLP	924,984	On	On	47.68	45.78
SS2D (matched recurrent)	924,224	On	On	**36.16**	**18.24**
SS2D (zero-history continuation)	924,224	Zero	Zero	41.02	22.05
		Zero	On	**40.79**	**14.75**
SS2D main model	924,224	On	Zero	43.53	26.21
		On	On	**41.15**	**22.52**

State propagation with adapters retained. For each zero-history and selected main SS2D checkpoint, we compare propagation on/off using identical weights and inference schedules. Restoring propagation for the zero-history step-400 checkpoint lowers ATE by 0.56%/33.12% on test_2/test_3. For the main checkpoint, ATE is 41.1526/22.5205 m with propagation versus 43.5276/26.2132 m when the outgoing inter-window state is zeroed, reductions of 5.46%/14.09%. All SS2D, residual, and candidate-write computations remain active. Segment-initial depth and pose are bitwise identical within each pair; only subsequent windows can read nonzero history.

5 Conclusion

Mamba-VGGT connects frozen VGGT windows through joint SS2D state–token adaptation, coupling residual feature updates with fixed-capacity spatial memory. Matched recurrent comparisons favor SS2D over the tested similarly sized pointwise MLP, while fixed-weight state interventions support the contribution of propagated history. With the reported decay and reset schedules, the adapter achieves lower ATE than LoGeR on the evaluated Ciampino inputs exceeding 18,000 frames. On Campus at 10,000 frames, it also reduces end-to-end runtime and peak allocated GPU memory relative to LoGeR under the stated implementations. These results highlight the potential of lightweight adaptation for long-sequence geometric reconstruction.

AI USE STATEMENT

Generative AI tools were used to assist with language editing, manuscript organization, preparation of portions of the LaTeX source, implementation of evaluation adapters, and experiment orchestration. They did not train the models or generate experimental measurements. Evaluations used the models trained in this work and the released baseline checkpoints under the stated configurations. Reported metrics and derived comparisons were computed from these evaluations, not estimated or fabricated by generative AI. The authors reviewed the adapters, commands, logs, aggregate data, text, and citations, and take responsibility for the final content and artifacts.

REPRODUCIBILITY STATEMENT

Sections 3 and 4 describe the method and experimental setup. The appendices provide additional details on training, inference settings, data sampling, alignment and evaluation protocols, ablation controls, and resource-measurement procedures. These descriptions clarify the configurations and evaluation conditions used in the reported experiments. Our code and checkpoints are not publicly available at submission; we plan to release them after submission or acceptance.

ETHICS STATEMENT

This work uses existing research datasets and does not introduce a new dataset or collect human-subject data. As with other reconstruction systems, applying the method to imagery of private spaces could reveal sensitive spatial information. Any deployment should respect applicable consent, privacy, and dataset-license requirements.

REFERENCES

Dejan Azinovic, Ricardo Martin-Brualla, Dan B. Goldman, Matthias Nießner, and Justus Thies. Neural RGB-D surface reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6280–6291, 2022.

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

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

Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas A. Funkhouser, and Matthias Nießner. ScanNet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2432–2443, 2017.

Afshin Dehghan, Gilad Baruch, Zhuoyuan Chen, Yuri Feigin, Peter Fu, Thomas Gebauer, Daniel Kurz, Tal Dimry, Brandon Joffe, Arik Schwartz, and Elad Shulman. ARKitScenes: A diverse real-world dataset for 3d indoor scene understanding using mobile RGB-D data. In *NeurIPS Datasets and Benchmarks Track*, 2021.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. In *Conference on Language Modeling*, 2024.

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

Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3d with MASt3R. In *European Conference on Computer Vision*, pp. 71–91, 2024.

Yiyi Liao, Jun Xie, and Andreas Geiger. KITTI-360: A novel dataset and benchmarks for urban scene understanding in 2D and 3D. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45 (3):3292–3310, 2023.

Lu Ling, Yichen Sheng, Zhi Tu, Wentian Zhao, Cheng Xin, Kun Wan, Lantao Yu, Qianyu Guo, Zixun Yu, Yawen Lu, Xuanmao Li, Xingpeng Sun, Rohan Ashok, Aniruddha Mukherjee, Hao Kang, Xiangrui Kong, Gang Hua, Tianyi Zhang, Bedrich Benes, and Aniket Bera. DL3DV-10K: A large-scale scene dataset for deep learning-based 3d vision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22160–22169, 2024.

Yue Liu, Yunjie Tian, Yuzhong Zhao, Hongtian Yu, Lingxi Xie, Yaowei Wang, Qixiang Ye, Jianbin Jiao, and Yunfan Liu. VMamba: Visual state space model. In *Advances in Neural Information Processing Systems*, 2024.

Jeremy Reizenstein, Roman Shapovalov, Philipp Henzler, Luca Sbordone, Patrick Labatut, and David Novotný. Common objects in 3d: Large-scale learning and evaluation of real-life 3d category reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10881–10891, 2021.

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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2538–2547, 2017.

Yu Sun, Xinhao Li, Karan Dalal, Jiarui Xu, Arjun Vikram, Genghan Zhang, Yann Dubois, Xinlei Chen, Xiaolong Wang, and Sanmi Koyejo. Learning to (learn at test time): RNNs with expressive hidden states. In *International Conference on Machine Learning*, 2025.

Hengyi Wang and Lourdes Agapito. 3d reconstruction with spatial memory. In *International Conference on 3D Vision*, 2025.

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

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

Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jérôme Revaud. DUSt3R: Geometric 3d vision made easy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20697–20709, 2024.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22378–22389, 2024.

Junyi Yang, Alexander Sax, Kevin J. Liang, Mikael Hénaff, Hao Tang, Ang Cao, Joyce Chai, Franziska Meier, and Matt Feiszli. Fast3R: Towards 3d reconstruction of 1000+ images in one forward pass. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1787–1796, 2020.

Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. ScanNet++: A high-fidelity dataset of 3d indoor scenes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 12–22, 2023.

Junyi Zhang, Charles Herrmann, Junhwa Hur, Chen Sun, Ming-Hsuan Yang, Forrester Cole, Trevor Darrell, and Deqing Sun. LoGeR: Long-context geometric reconstruction with hybrid memory. *arXiv preprint arXiv:2603.03269*, 2026.

Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. *ACM Transactions on Graphics*, 37(4):65:1–65:12, 2018.

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

A Training Details and Evaluation Protocols

A.1 Training Data, Optimization, and Supervision

Training lineage and scope. Main-model training starts from the pretrained VGGT checkpoint, with the backbone and camera, depth, point, and track heads frozen. The added adaptation modules are initialized afresh using the policy in Section 3.2, including zero-initialized token-residual and state-write output projections. Only these added components are trained; they contain 7,915,008 parameters, approximately 0.63% of the 1,256,537,516-parameter VGGT backbone/head model. The main model has two training stages: 32-frame (32f) adaptation followed by 64-frame (64f) continuation. The 32f update-3,400 checkpoint initializes the 2,000-update 64f stage, from which update 1,800 is selected for quality evaluation. The two stages share the training recipe below—data mixture, preprocessing, sample batch size, optimizer, losses, and window/state handling— with only the sampled sequence length changing from 32 to 64 frames; their update counts are specified above. All three ablation continuations branch from the same 32f checkpoint, not the final 64f model. They inherit compatible backbone, head, token-residual, and state-write parameters, but initialize their mixers afresh: SS2D in the zero-history and matched recurrent branches, and the replacement MLP in the recurrent MLP branch. The recurrent SS2D and MLP branches share writer updates, data, optimization, recurrent state handling, and the validation checkpoint-selection rule. The zero-history branch fixes its inherited writer and suppresses inter-window state during continuation, while other active adapter parameters are optimized. Section 4.5 and Appendix D.2 distinguish these protocols. Reproduction manifests record the model identity and exact inputs.

Data, batch size, and resolution. Both stages mix CO3D (Reizenstein et al., 2021), ARKitScenes (Dehghan et al., 2021), WildRGB-D (Xia et al., 2024), BlendedMVS (Yao et al., 2020), DL3DV-10K (Ling et al., 2024), and ScanNet++ (Yeshwanth et al., 2023), with a nominal global quota of 600 samples per source per epoch. Both stages use three distributed workers, one sample per GPU per microbatch, and two-step gradient accumulation. The effective optimization batch is six samples: 192 sampled frames in 32f and 384 in 64f, before overlap repetitions. Inputs have nominal width 518 and patch size 14. Training heights are 210, 266, 294, 336, 392, and 518; validation uses 294, 392, and 518. All frames in a microbatch share the sampled aspect ratio. This is not one fixed 518-by-392 training resolution.

Optimizer and schedule. AdamW uses a learning rate of 10^{-5} for memory-module parameters, weight decay 0.01, $(\beta_1, \beta_2) = (0.9, 0.999)$, and $\epsilon = 10^{-8}$, as recorded in the checkpoint optimizer. The trainer has no learning-rate scheduler or learning-rate warm-up; the learning rates remain constant. CUDA autocast uses BF16, with FP32 parameter storage. Frozen-head/backbone learning-rate entries in the configuration do not create active optimizer groups. Validation and checkpointing occur every 200 successful updates. Early stopping monitors the equal-source mean native validation objective with patience ten, minimum improvement 0.001, and a 600-update early-stopping warm-up (not an LR warm-up). The 64f stage reaches 2,000 updates without early stopping; its selected checkpoint is update 1,800.

Temporal differentiation. Both stages use 16-frame windows with four-frame overlap, and each sample starts from zero memory. A 64f sample spans five windows. In both stages, state is not detached between windows: BPTT spans the full 32f or 64f sample, but not separate samples or accumulated microbatches. Writes commit a frame at its first appearance; repeated overlap frames are not written again. This permits a repeated frame to read a state containing its earlier write. Overlap-consistency teachers from the previous window and the fitted alignment are detached, which is distinct from detaching recurrent state.

Losses and weights. GT is re-anchored to each window's first camera and scale-normalized, with at least 100 valid points for scale estimation. The full objective is

$$\mathcal{L} = \mathcal{L}_{\text{cam}} + \mathcal{L}_{\text{depth}} + 0.35\mathcal{L}_{\text{point}} + 0.10\mathcal{L}_{\text{overlap}}. \tag{11}$$

The coefficients below are applied to the implementation's reduced loss components (not unnormalized sums over pixels or pairs):

- Camera: translation 0.1, rotation 1.0, focal length 0.05; relative rotation 1.0 on adjacent pairs and relative translation 0.5 on all pairs, with translation normalization $\epsilon = 10^{-6}$.
- Depth: confidence 1.0, regression 1.0, scale-invariant term 1.5, gradient term 0.0. Point: confidence 1.0, regression 1.0, gradient term 0.0; point-normal loss is disabled. The outer point coefficient is 0.35.
- Shared-Sim(3) overlap: camera centers 1.0, rotations 1.0, points 1.0, depth 0.5, multiplied by the outer coefficient 0.10. Residuals use Smooth-L1 with $\beta = 0.1$. The transform and previous-window teacher are stop-gradient. Differentiable cross-head consistency is disabled, so its configured default coefficient does not contribute. Memory-read-content regularization has coefficient 0.0.

All non-ablation Mamba-VGGT results use the same pretrained adaptation checkpoint.

Training-resource audit. For the 64-view continuation, logged main-rank interval times sum to 20,061.88 s through update 1,800 (5.573 h), approximately 16.72 GPU-hours with three workers. Through update 2,000 the sum is 22,289.42 s (6.192 h), approximately 18.57 GPU-hours. These intervals include intervening validation/checkpoint overhead but exclude startup and operations after the final timing line; they are not a complete end-to-end training bill. The maximum logged main-rank PyTorch allocated peak is 128.95 GiB (138.459 GB). It is a cumulative observed peak, not an all-rank maximum or reserved-memory peak. Earlier 32-frame adaptation cost is not included and has not been reconstructed.

A.2 Evaluation Data and Sampling

Short-sequence evaluations use a nominal 32-frame setting. KITTI-360 and RE10K sequences contain 32 frames; ETH3D uses up to 32 matched RGB-depth-camera views, retaining 31 views for kicker and 15 for meadow without repetition. The exact frame identities and counts are listed in the supplementary protocol manifests. The retained reconstruction and depth tables cover ETH3D and KITTI-360; the pose table covers ETH3D and RE10K. For KITTI, four sequences use valid-frame indices $0, 2, \ldots, 62$. We retain the fixed 32-frame sampling manifests. The medium-length KITTI experiment uses 300 RGB-adaptive samples from each of test_0–test_3: all 933 source frames for test_0 and the first 1000 for each other block. Sampling uses adjacent-frame Farnebäck flow on 160-pixel-wide grayscale images. Interval weights combine a uniform term and mean-normalized median flow magnitude (clipped at 10 pixels), each weighted by 0.5. Equal cumulative weight sampling retains both endpoints and enforces unique increasing indices; GT motion and model outputs are not used for selection. Reconstruction and pose share inputs and predictions, and metrics are averaged equally across blocks. The NRGBD depth comparison in Table 2 uses the nominal 300-frame setting, yielding 304 input frames by integer-stride sampling over the full trajectory. The exact input list is provided in the supplementary material. For the long trajectories, 1000 or 2000 observations are uniformly sampled from the first 2500 valid observations of each of test2 and test3. These settings share endpoints but differ in input density.

RE10K reference scale and pose-error units. RealEstate10K provides camera trajectories reconstructed from monocular video using SLAM and bundle adjustment. Its construction adopts a scene-depth normalization convention to address per-clip scale ambiguity, rather than establishing independently measured metric scale (Zhou et al., 2018). We invert the released world-to-camera extrinsics and retain their native translation scale without additional metric calibration. ATE is computed after a single global Sim(3) alignment to the reference camera centers and is therefore reported in native dataset units, not meters. All methods use identical reference trajectories and sampled frames, with equal averaging across sequences. These results support within-protocol comparisons but are not interpreted as physical distances directly comparable to metric ATE on ETH3D, KITTI-360, or VBR. Rotation and translation-direction errors remain in degrees.

A.3 Baselines, Preprocessing, and Inference

We compare with pretrained VGGT executed independently in matched 16/4 windows (VGGT-W16), the official Spann3R v1.01, CUT3R, TTT3R, StreamVGGT with its full sequence KV cache, and ZipMap-W12; the medium- and long-sequence comparisons also include LoGeR with SWA under

the window configuration below. ZipMap-W12 uses the released online/AR checkpoint and native 12-frame TTT windows, carrying fast-weight state across windows (not 12 windows of history). Our adapter loads the full image sequence in one call; this preserves the internal online update order but is not a frame-at-a-time peak-memory benchmark. TTT3R uses the released CUT3R 512-DPT checkpoint with its official test-time update mode. Every method receives the same sampled frame identities and is scored by the same evaluator. We first apply the geometric preprocessing required by each dataset to produce canonical RGB inputs, including ETH3D image undistortion and indoor RGB/depth registration. Each external method then retains its native image transform: CUT3R and TTT3R use the DUSt3R long-side-512 resize and center crop, while Spann3R uses its official 224×224 center crop. Before scoring, every prediction is mapped back to the evaluator's canonical pixel grid. No method receives ground-truth camera calibration. CUT3R and TTT3R use their pose heads and self-view point maps to estimate focal length; Spann3R uses its official shared first-view focal estimate and per-frame PnP-RANSAC. Degenerate focal estimates are repaired without ground truth, using the sequence median of valid estimates when available and otherwise falling back to a pinhole camera with a 60° field of view. For the length studies, Mamba-VGGT uses four taps and a leak5 update at taps 16 and 22: the first five writes are unchanged, and subsequent updates use $M_{t+1} = 0.8M_t + \Delta M_t$. This is exponential forgetting, not a hard five-window FIFO. Both Mamba-VGGT and VGGT-W16 use matched 16/4 windows and fixed-scale SE(3) stitching for KITTI reconstruction and pose, without additional segment resets in the 300-frame experiments. On the separate 1000/2000-frame KITTI runs, both Mamba-VGGT and VGGT reset their segment execution every 300 new frames with four overlap frames; Mamba-VGGT reinitializes all four memory states and the leak warm-up. TTT3R uses its official reset100 state/pose-memory reset and cumulative camera-base composition throughout the length studies. LoGeR uses the released W32/O3 checkpoint with matched 16/4 evaluation windows, TTT and SWA, and a 255,000-pixel full-image budget; the 300-frame KITTI and Kitchen depth runs have no additional segment resets. Other baselines retain their native inference interfaces. *Adapter off* uses the original VGGT aggregation path. *State propagation off* instead keeps feature adaptation active while resetting the state input to zero for every window, as defined in Section 3.2.

A.4 Sparse KITTI Reconstruction and Evaluation Alignment

For KITTI reconstruction, GT points and cameras share the first rectified-camera frame. Camera-to-world poses are checked against rig poses and calibration. Each method reconstructs the same LiDAR rays with estimated intrinsics and poses, bilinear predicted-depth sampling, and common valid support. GT camera depth is restricted to $(0.1, 80]$ m, with nearest-depth z-buffering on native pixels. A deterministic approximately 20% ray subset fits one global positive-scale Sim(3); the disjoint remaining rays are scored on a shared 0.1 m voxel grid. This GT-based transform is used only for evaluation, not window stitching. We report Acc., Comp., and their mean CD. Dynamic objects are not removed and LiDAR scans are not motion-deskewed. This sparse observed-support protocol is kept separate from dense indoor reconstruction; NC is not reported for this experiment.

B Short-Context Results and Visualizations

Table B1: Nominal 32-frame inputs; actual per-sequence counts are specified in Appendix A.2. KITTI-360 uses every second valid observation. Methods share inputs and evaluation support within each panel. Distances are meters after global Sim(3); NC is higher-is-better. KITTI reports CD rather than NC on sparse LiDAR support. Bold marks the best value before rounding.

Method	ETH3D (Schöps et al., 2017) Acc.↓	Comp.↓	NC↑	KITTI-360 (Liao et al., 2023) Acc.↓	Comp.↓	CD↓
VGGT-W16 (Wang et al., 2025a)	**0.9609**	0.5629	0.7912	**0.3624**	0.6238	0.4931
Spann3R (Wang & Agapito, 2025)	24.72	1.458	0.6747	0.6603	0.6710	0.6656
CUT3R (Wang et al., 2025b)	1.441	0.9286	0.6793	0.5339	1.527	1.030
TTT3R (Chen et al., 2025)	1.519	0.8946	0.7026	0.4823	1.068	0.7754
StreamVGGT (Zhuo et al., 2025)	1.316	1.038	0.6913	0.7320	0.5750	0.6535
ZipMap-W12 (Jin et al., 2026)	1.731	0.6192	0.7447	0.3714	0.5500	0.4607
Mamba-VGGT	0.9652	**0.3648**	**0.8186**	0.3764	**0.4880**	**0.4322**

Table B2: Nominal 32-frame inputs; actual per-sequence counts are specified in Appendix A.2. AbsRel and $\delta < 1.25$ use one median scale per sequence; KITTI-360 evaluates against sparse LiDAR. Bold marks the best value before rounding.

Method	ETH3D (Schöps et al., 2017) AbsRel↓	δ_1 ↑	KITTI-360 (Liao et al., 2023) AbsRel↓	δ_1 ↑
VGGT-W16 (Wang et al., 2025a)	0.04480	0.9769	0.09460	0.9181
Spann3R (Wang & Agapito, 2025)	0.1489	0.8338	0.2112	0.6786
CUT3R (Wang et al., 2025b)	0.08340	0.9276	0.1798	0.7052
TTT3R (Chen et al., 2025)	0.07130	0.9390	0.1666	0.7564
StreamVGGT (Zhuo et al., 2025)	0.1245	0.8681	0.1479	0.7894
ZipMap-W12 (Jin et al., 2026)	0.04680	**0.9942**	0.08690	0.9072
Mamba-VGGT	**0.03460**	0.9899	**0.08420**	**0.9343**

Table B3: Nominal 32-frame inputs; actual per-sequence counts are specified in Appendix A.2. ATE uses meters on ETH3D and native dataset units on RE10K; RPE_t is the translation-direction angular error and RPE_r is the rotation geodesic error, both in degrees. Bold marks the best value before rounding.

Method	ETH3D (Schöps et al., 2017) ATE↓	RPE_t ↓	RPE_r ↓	RE10K (Zhou et al., 2018) ATE↓	RPE_t ↓	RPE_r ↓
VGGT-W16 (Wang et al., 2025a)	1.730	17.28	2.381	0.02120	**8.374**	0.1352
Spann3R (Wang & Agapito, 2025)	2.474	29.79	10.87	0.08960	30.98	0.4864
CUT3R (Wang et al., 2025b)	2.407	21.32	7.339	0.04740	13.78	0.1755
TTT3R (Chen et al., 2025)	1.859	32.32	9.344	0.04930	10.29	0.1760
StreamVGGT (Zhuo et al., 2025)	3.211	23.96	7.199	0.1018	10.24	0.1637
ZipMap-W12 (Jin et al., 2026)	2.538	33.30	4.073	0.04280	17.30	0.2280
Mamba-VGGT	**0.5457**	**7.629**	**1.843**	**0.01670**	8.693	**0.1327**

Reconstruction. On ETH3D, Mamba-VGGT achieves the lowest Comp. and highest displayed NC, with the second-lowest Acc.; Comp. decreases from 0.5629 m to 0.3648 m relative to VGGT-W16 (35.2%). On KITTI-360, Mamba-VGGT gives the lowest Comp. and CD: CD is 0.4322 m versus 0.4931 m for VGGT-W16 and 0.7754 m for TTT3R, reductions of 12.4% and 44.3%. VGGT-W16 and ZipMap-W12 have lower Acc., so this is a coverage–accuracy trade-off.

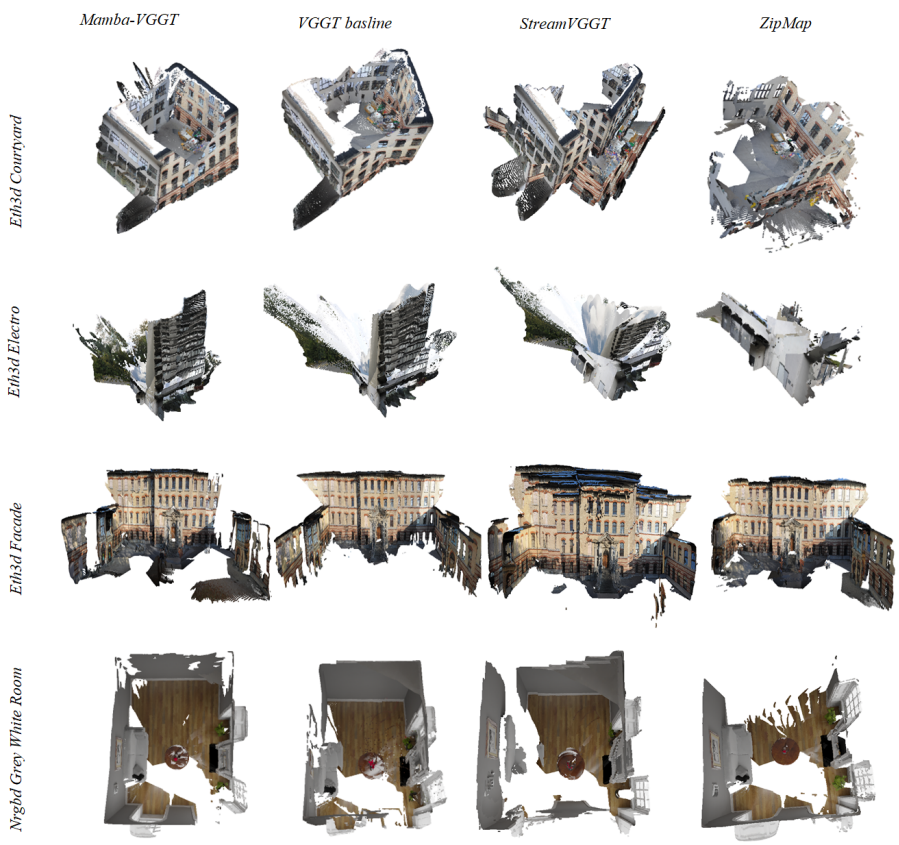


Figure B1: Qualitative reconstruction examples on ETH3D *courtyard*, *electro*, and *facade*, and NRGBD *grey-white room*. Columns show Mamba-VGGT, VGGT baseline, StreamVGGT, and ZipMap. These illustrative renderings show differences in visible structures, fragmentation, and layout.

Qualitative reconstruction. Figure B1 shows ETH3D and NRGBD renderings, illustrating differences in visible surfaces, fragmentation, and global layout.

Depth and camera pose. Mamba-VGGT improves all four retained depth entries relative to VGGT-W16, and all four rank among the best two displayed values. ETH3D AbsRel decreases by 22.8%, and Mamba-VGGT gives the lowest AbsRel and highest $\delta < 1.25$ on KITTI-360. All six retained pose entries rank first or second: five are first and one is second. ETH3D ATE decreases from 1.7298 m to 0.5457 m (68.5%). On Re10K, Mamba-VGGT gives the lowest ATE and rotation error.

C Additional Medium- and Long-Sequence Results

C.1 Four-Block Medium-Length Visualizations

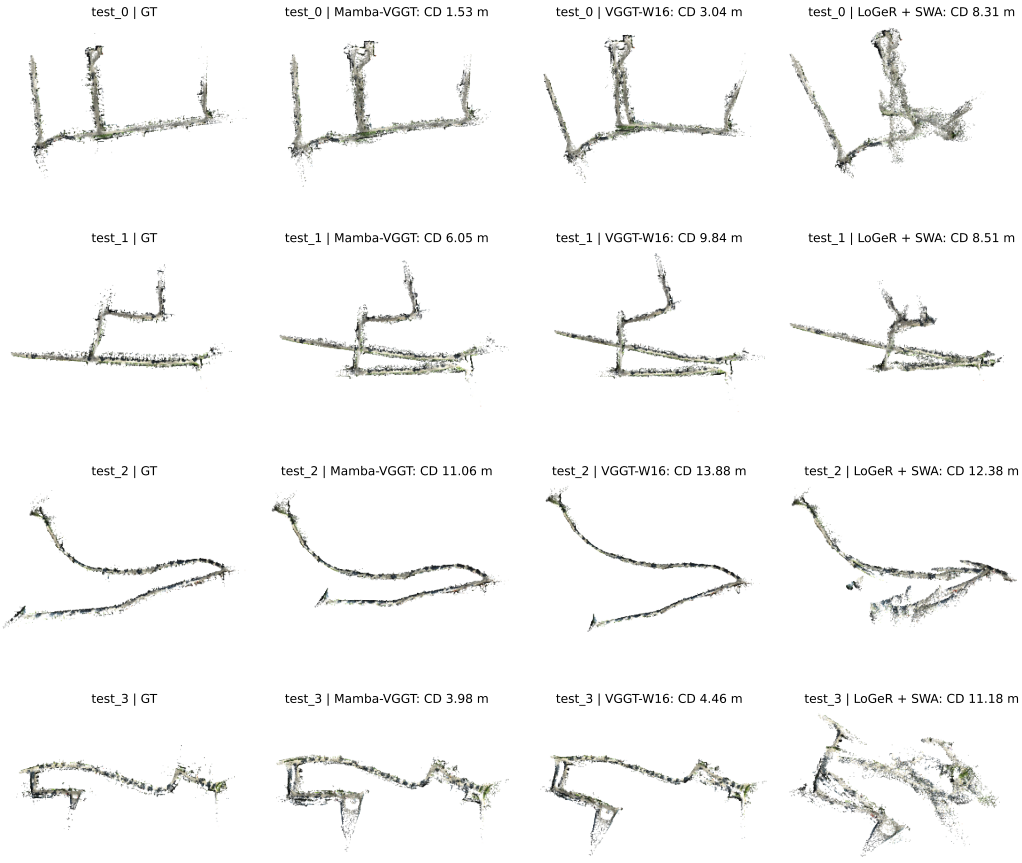


Figure C1: Four KITTI-360 blocks, 300 RGB-adaptive samples each. Rows are test_0–test_3; columns show GT, the main model with normal state propagation, VGGT-W16, and LoGeR + SWA. Each row uses the same deterministic sample of held-out common LiDAR observations and the reconstruction transforms used for Table 2. Labels show per-block CD, not the four-block mean. RGB colors, a GT-derived plane, display scale and limits are shared within each row, with axes removed; there is no ICP, smoothing or clipping. Test_2's unsupported frame contributes no points. These panels show observed-support geometry, not dense full-scene coverage. The test_0, test_2, and test_3 rows are reproduced in Figure 4.

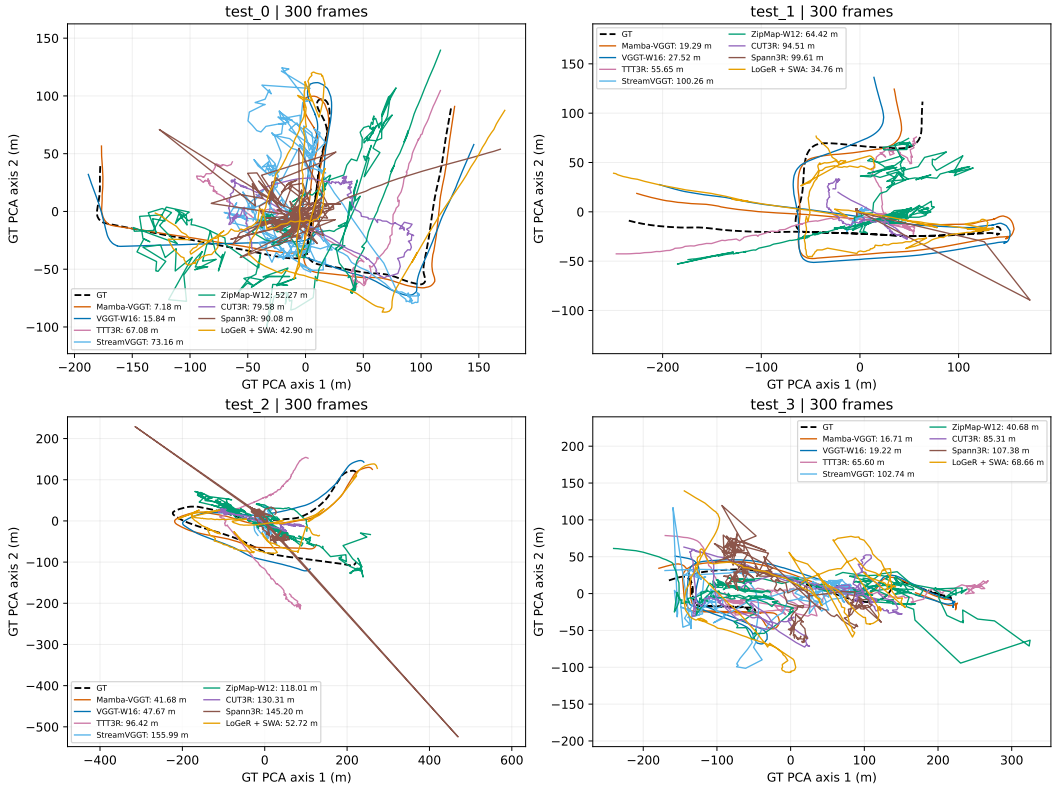


Figure C2: Camera trajectories for all four KITTI-360 blocks in Table 2, with 300 sampled poses per method and block. Black denotes GT; Mamba-VGGT uses the selected main checkpoint with normal state propagation. Each complete trajectory receives one global Sim(3) evaluation alignment. All eight methods share the GT-derived plane and metric axes within each panel, without smoothing, clipping or local GT correction. Legends report per-block ATE; the table reports their equal-weight four-block mean.

C.2 Long-Sequence Trajectories and Additional Point Clouds

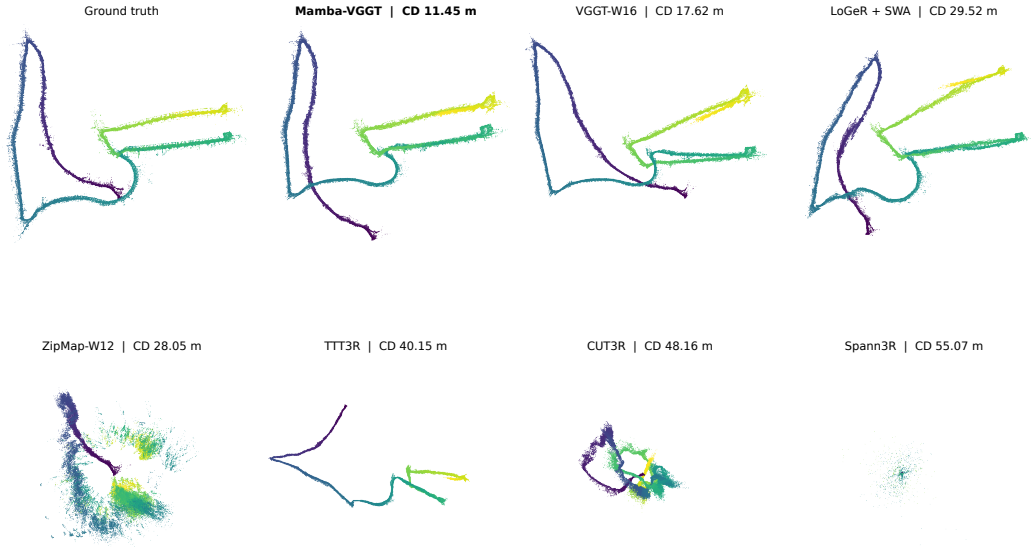


Figure C3: KITTI-360 test_2 reconstruction with 2000 frames, showing all successful methods. Identical held-out observations, coloring, orthographic view and display scale are shared across panels; coordinate axes are removed. StreamVGGT is omitted only because its complete run fails with OOM.

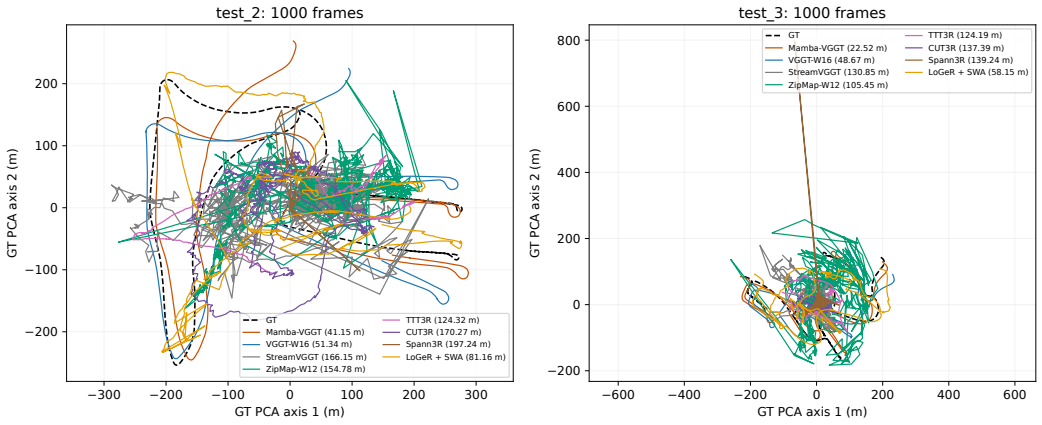


Figure C4: All successful baselines on test_2 and test_3 at 1000 frames. Legends give full-sequence ATE. Each panel uses a shared GT-derived plane; all trajectory points are shown without smoothing or clipping.

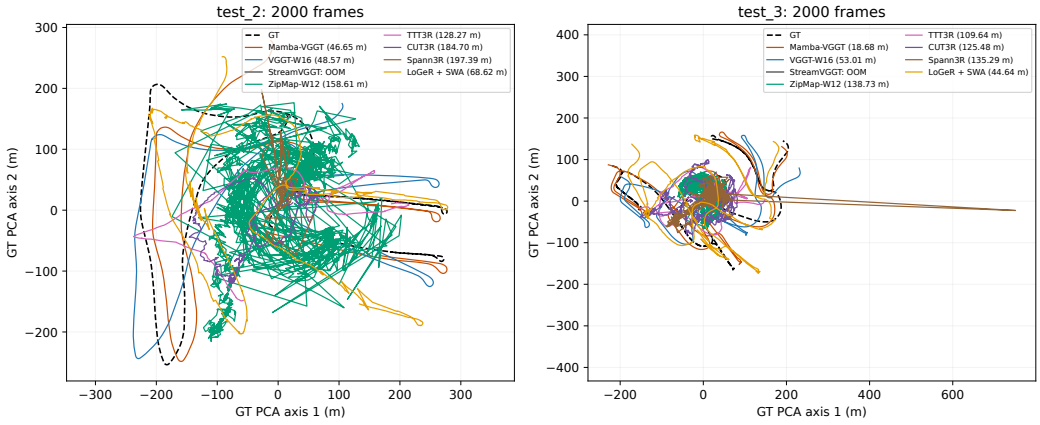


Figure C5: The same two cases at 2000 frames. StreamVGGT has no complete prediction because of OOM; successful trajectories correspond to Table 3.

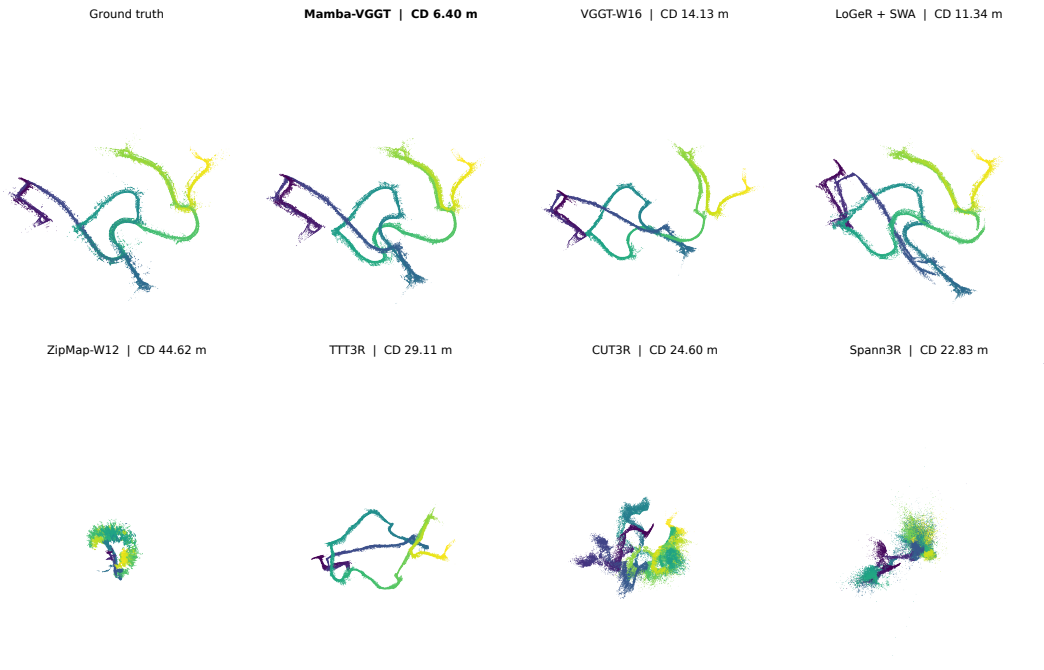


Figure C6: Test_3 at 2000 frames, using the same held-out-support visualization protocol as Figure 5. Color denotes source-frame index; labels report per-method CD. No output is fabricated for OOM.

D Adapter Details and Resource Measurements

D.1 SS2D Implementation

The main equations use a single joint mixer Φ_ℓ. The following gives its expanded spatial form; none of the scan hidden states are persisted between video windows. The state mixer follows Mamba's selective state-space model and the two-dimensional selective scan of VMamba (Gu & Dao, 2024; Liu et al., 2024). For a single spatial input $J \in \mathbb{R}^{H_p \times W_p \times (C+K)}$, the input projection produces content and gate branches,

$$[U, Z] = W_{\text{in}} J, \qquad \widetilde{U} = \text{SiLU}(\text{DWConv}_{3\times 3}(U)), \qquad G = \text{SiLU}(Z). \tag{12}$$

CrossScan unfolds $\widetilde{U}$ along four paths: row-major, column-major, and their reverse orders. At position p in direction d, the current input predicts the selective step and the standard SSM input and output coefficients $(\Delta_{d,p}, B_{d,p}, C_{d,p})$. With $A_d = -\exp(A_d^{\log})$, the discretized recurrence is

$$h_{d,p} = \exp(\Delta_{d,p} A_d) \odot h_{d,p-1} + \Delta_{d,p} B_{d,p} \odot u_{d,p}, \tag{13}$$

$$y_{d,p} = C_{d,p} h_{d,p} + D_d \odot u_{d,p}. \tag{14}$$

The feature and SSM-state axes are omitted for clarity. Here $B_{d,p}$ and $C_{d,p}$ are position-dependent SSM coefficients, not the batch dimension or cross-window state variables. The transition, input coupling, and readout are therefore conditioned on current content. As in VMamba, CrossMerge restores the four outputs to a common patch-grid orientation and sums them. The complete mixer is

$$\Phi(J) = W_{\text{out}}\big(\text{LN}\big(\text{CrossMerge}(\{y_d\}_{d=1}^4)\big) \odot G\big). \tag{15}$$

After spatial mixing, the joint output is split into memory and token channels,

$$[Y_{m,s}^{M,\ell}, Y_{m,s}^{X,\ell}] = \text{Split}_{K,C}\big(\Phi_\ell(J_{m,s}^\ell)\big). \tag{16}$$

In our implementation, $C = 1024$, $K = 64$, the internal SS2D width is 256, and each one-dimensional selective scan has state dimension 16. The same Φ is applied independently to all BS_m frame grids, using VMamba's CrossScan, CrossMerge, and grouped selective-scan CUDA operators. The SSM hidden state h is transient within one frame-grid call and is reset between calls. Only the explicit activation M persists across windows.

The joint mixer is illustrated in Figure 2 in the main text.

D.2 Ablation Initialization and Controls

The three continuation variants in Table 6 start from the 32f update-3400 checkpoint, not the final main model. In all three, compatible backbone, prediction-head, token-residual, and state-write parameters are inherited, but the mixer is initialized afresh. In the zero-history branch, the writer retains the learned 32f projections and biases and is frozen; window input/output states are forced to zero while the remaining active adapter parameters are optimized. Continuation step 400 is selected by validation. Its state on/off pair uses the same RTX 5090 and reference SS2D implementation. For the selected main checkpoint (step 1800), both state conditions run on H200. Within each pair, frames, weights, window plan, leak5, reset300, and fixed-scale stitching are identical. Each reset segment's first-window depth and pose are bitwise equal; subsequent windows differ. Figure D1 shows the zero-history pair with earlier baseline runs, not the main VBR or long-sequence comparison.

The MLP is linear $1088 \to 424$, SiLU, LayerNorm, and linear $424 \to 1088$ at each position, with no spatial convolution or selective scan. Its newly initialized mixer is combined with the inherited compatible components; token-residual and state-write projections are not re-zeroed. This branch retains window-level recurrent training and uses the stated data mixture, losses, and 2000-update continuation schedule, with validation-selected step 1400. The matched recurrent SS2D branch likewise initializes its mixer afresh, retains the same inherited components, and uses the same recurrent training, writer updates, data, losses, optimization, continuation schedule, and validation checkpoint-selection rule as the MLP. It achieves ATE 36.16/18.24 m with state propagation on; no state-off measurement is reported for this branch. Neither mixer in this matched pair inherits its own 32f pretraining. The zero-history continuation is a separate state-propagation control, not the SS2D comparator for the MLP. The selected main checkpoint retains its trained SS2D mixer and is reported as a separate reference in Table 6.

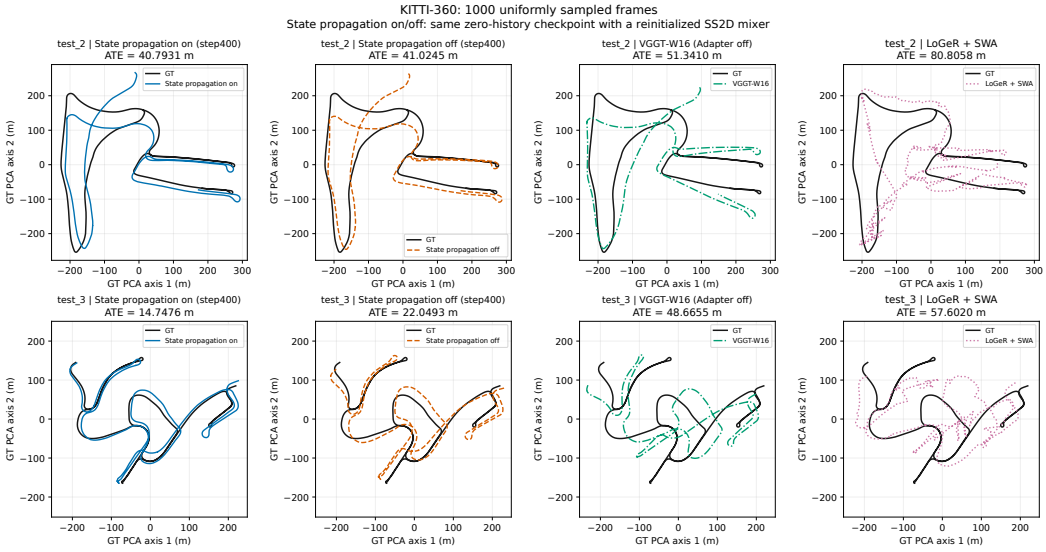


Figure D1: KITTI-360 test_2 (top) and test_3 (bottom), 1000 frames each. Columns: state propagation on/off after the same zero-history continuation with a reinitialized SS2D mixer, VGGT-W16 (Adapter off), and the earlier LoGeR + SWA baseline run. GT is black. Each trajectory uses one global Sim(3); labels give 3D ATE. Each row shares a GT-derived plane, equal axis scales, and display limits, without smoothing, local GT correction, or trajectory clipping.

D.3 INFERENCE FOOTPRINT

We report a storage–quality trade-off, not a universal speed or peak-memory advantage. Table D1 reports repeated measurements with the quality-evaluation configuration. All methods run sequentially on the same H200, using identical 32 uniformly sampled ETH3D frames at 518×350, FP32 parameter storage, and BF16 autocast (native heads may internally disable autocast). Each method has one unmeasured warm-up followed by five fresh-state repetitions. CUDA is synchronized around the measured call; no other GPU task is co-scheduled. The table gives population mean and standard deviation of elapsed time, $\mathrm{FPS} = 5 \times 32 / \sum_{r=1}^{5} t_r$, and the maximum allocated peak across the five runs. FPS counts unique input frames, not repeated overlap computations; it is not an arithmetic mean of per-run FPS.

Input images are decoded on CPU before timing. The timed region includes host-to-device staging, native forward execution, and copying returned predictions to CPU. It excludes checkpoint loading, disk I/O, external stitching, scoring, and visualization. Windowed methods stage active windows; ZipMap and StreamVGGT stage the complete 32-frame input for their native interfaces. Model-specific attention/TTT/SS2D kernels and native output payloads remain unchanged. PyTorch SDPA flash, memory-efficient, and math backends are enabled, but enabled flags do not establish that identical kernels were selected by every method. No fused `selective_scan_cuda_oflex` extension was available.

Table D1: Repeated 32-frame inference measurements on ETH3D (Schöps et al., 2017). Parameters count loaded inference weights; persistent state is retained tensor payload in decimal MB; peak is PyTorch allocated memory in decimal GB; time is mean ± population SD over five fresh-state runs. The matched VGGT wrapper retains inactive spatial-memory weights, whereas the native wrapper loads only VGGT.

Method	Params (B)	State (MB)	Peak (GB)	Time (s)	FPS
VGGT-W16 (Wang et al., 2025a) (matched wrapper)	1.264	0.000	9.215	0.8250 ± 0.02200	38.79
VGGT-W16 (Wang et al., 2025a) (native wrapper)	1.257	0.000	11.62	0.8210 ± 0.003000	38.98
StreamVGGT (Zhuo et al., 2025)	1.257	2934	12.57	2.740 ± 0.05600	11.68
ZipMap-W12 (Jin et al., 2026)	1.168	604.0	9.982	0.5750 ± 0.001000	55.68
Mamba-VGGT	1.264	0.9470	9.232	0.8330 ± 0.006000	38.41

The added adaptation components total 7.915M parameters (0.63% of VGGT). Their 0.947 MB persistent state is approximately $638\times$ smaller than ZipMap's fast-weight payload and over $3{,}000\times$ smaller than the 32-frame StreamVGGT cache. These ratios describe persistent state, not total GPU memory. Mamba-VGGT uses 9.232 GB allocated peak versus 9.215 GB for the matched VGGT wrapper and 11.616 GB for native VGGT. At 38.41 FPS, Mamba-VGGT is slightly below the matched VGGT wrapper (38.79 FPS) and below ZipMap-W12 (55.68 FPS). Feature retention, tensor lifetime and native interface choices affect peak memory and throughput; bounded recurrent storage alone does not imply a speed advantage.

Training cost and its accounting scope are reported in Appendix A.1.

D.4 LEAK5 STATE STABILITY AND TERMINAL-WINDOW GEOMETRY

Leak5 acts only on taps 16/22. The first five writes retain the native update; subsequent writes use $M_{t+1} = 0.8M_t + \Delta M_t$, leaving taps 3/10 unchanged. This is exponential forgetting, not a strict five-window FIFO. We use two ScanNet sequences of 300 consecutive frames and examine the 24th complete window (zero-based input indices 276–291). This window is not selected by an error peak.

Table D2: Late-tap magnitudes at the fixed terminal window on ScanNet (Dai et al., 2017). State RMS measures incoming memory, and residual RMS measures the patch-feature residual to be injected. Residual/token is residual RMS divided by the original patch-token RMS, not a state-magnitude ratio.

Scene	Setting	Tap	State RMS	Residual RMS	Residual/token
scene0499_00	No leak5	16	6.837	0.9392	1.788
scene0499_00	No leak5	22	10.15	2.187	1.415
scene0499_00	Leak5	16	0.9253	0.1575	0.2998
scene0499_00	Leak5	22	1.939	0.8109	0.6354
scene0233_01	No leak5	16	6.867	0.9389	1.833
scene0233_01	No leak5	22	11.04	2.285	1.523
scene0233_01	Leak5	16	0.8711	0.1430	0.2792
scene0233_01	Leak5	22	2.006	0.8596	0.6617

Without leak5, both late-tap patch residuals exceed the original token RMS on both scenes. Leak5 suppresses these ratios and the incoming-state magnitude. For example, on scene0499_00, tap16 state RMS falls from 6.83731 to 0.92531 and tap22 from 10.15232 to 1.93870. Magnitude control alone does not establish geometric improvement, so we additionally measure local tasks and intervene on readouts.

Table D3: ScanNet (Dai et al., 2017) local geometry at the same window as Table D2. Depth uses one window-median scale; local ATE and point clouds use a window-level camera-center Sim(3) evaluation alignment. Clouds use predicted intrinsics, common valid GT support, fixed pixel sampling, and a common 2 cm voxel grid. Local CD is the mean of the two directional distances. These window-level scores are not full 300-frame metrics.

Scene	Setting	Local AbsRel↓	Local ATE (m)↓	Local CD (m)↓
scene0499_00	No leak5	0.06571	0.02658	0.2648
scene0499_00	Leak5	0.03048	0.01560	0.1308
scene0499_00	No leak5; target-window read muted	0.03005	0.01997	0.1694
scene0233_01	No leak5	0.05145	0.01342	0.1312
scene0233_01	Leak5	0.02543	0.01158	0.05778
scene0233_01	No leak5; target-window read muted	0.02208	0.01410	0.1027

Leak5 reduces terminal-window AbsRel by 53.61%/50.58% and CD by 50.61%/55.96% on the two scenes. The reported percentage changes refer to unrounded measurements; Tables D2 and D3 display four significant figures. To test whether late readouts cause harm, we keep all four incoming states and the input history identical and disable only tap16/22 patch residuals in the target window, retaining the other readouts and candidate-write computation. The preceding 23 windows are bitwise identical, as are the target-window incoming-state hashes. Without leak5, this intervention also improves local depth and CD, supporting the hypothesis that excessive late patch readouts can damage local geometry.

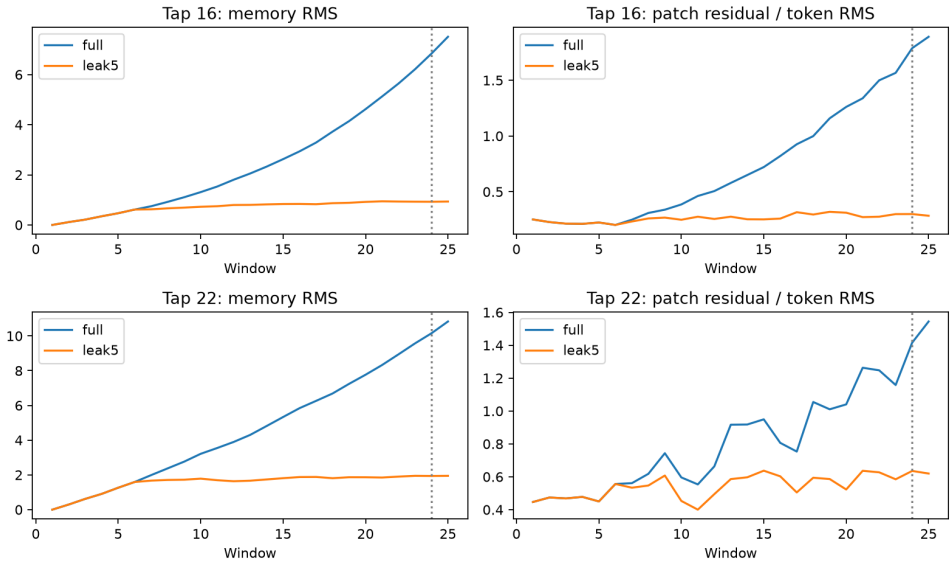


Figure D2: Tap16/22 state RMS and patch-residual/token ratios across windows on scene0499_00. The dashed line marks the fixed diagnostic window. “full” denotes history without leak5; “leak5” denotes history with leak5.

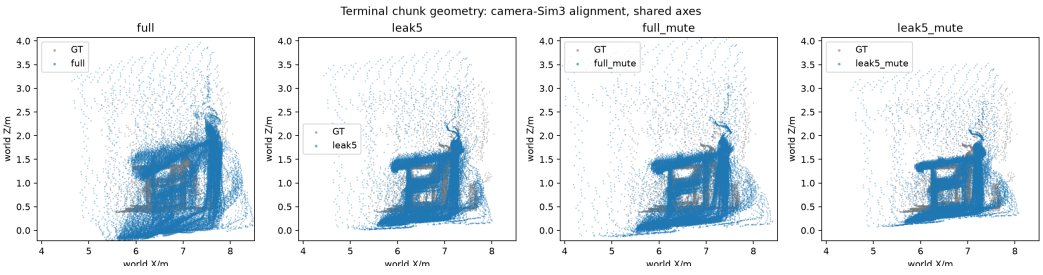


Figure D3: Fixed terminal-window point clouds and GT on scene0499_00, using shared display limits and one camera-based Sim(3) evaluation alignment per window. Gray is GT and blue is prediction. “full”/“leak5” denote history without/with leak5; “mute” disables only tap16/22 patch readouts in the target window. Scale alignment is used only for evaluation, not inference.

D.5 RESET300 AT THE 1000-FRAME TRACKING ENDPOINT

Table D4: Complete-trajectory evaluation at 1000 frames on KITTI-360 (Liao et al., 2023) test_2. Both settings use the same leak5 policy, 16/4 windows, and fixed-scale SE(3) stitching, with 1000 uniformly sampled frames from the same first 2500 valid observations. ATE uses one global Sim(3) alignment over all 1000 frames. "Endpoint" means the complete-trajectory score after processing 1000 frames, not the position error of the final camera alone. Values are rounded to four significant figures; percentage changes use unrounded measurements.

Setting	ATE (m)↓	RPE_r (°)↓	RPE_t (°)↓
Leak5, no reset	46.82	0.3667	6.849
Leak5, reset300	41.15	0.3386	6.892

Reset300 reduces complete-trajectory ATE by approximately 12.1% and rotation RPE by 7.68%. It reinitializes all four tap memories, the leak5 warm-up counter, and segment-window execution, retaining four boundary-overlap frames. This experiment therefore tests the deployment schedule as a whole, rather than an isolated state-zeroing operation.

E LONG-SEQUENCE REPRODUCTION AND EXECUTION DETAILS

Inputs and native execution. For each sequence and requested length $N \in \{1000, 2000\}$, frame indices are $\mathrm{round}(\mathrm{linspace}(0, 2499, N))$ within the first 2500 valid observations. Mamba-VGGT and VGGT use 16/4 windows and reset every 300 new frames with four boundary-overlap frames. LoGeR evaluates its released W32/O3 checkpoint with matched 16/4 windows, resets TTT at the same new-frame count, preserves SWA history, and retains three boundary-overlap frames with its native first-overlap rigid merge. The three-frame segment-boundary overlap is distinct from the four-frame overlap between windows inside a segment. No GT is used in segment registration. StreamVGGT retains its full native history and runs out of memory on both 2000-frame cases; those rows have no ATE or reconstruction estimate. CUT3R and Spann3R retain native state; TTT3R follows its official reset100/cumulative-camera-base protocol.

Reconstruction support. Long-run geometry starts with every fourth native projected LiDAR ray within $(0.1, 80]$ m, before intersecting finite positive bilinear predictions. Original ray indices divisible by five form the fitting subset; remaining indices are scored. The eight-method intersections at 1000 frames cover 36.77%/37.16% of prefiltered rays on test_2/test_3; the seven-method intersections at 2000 frames cover 36.83%/37.12%. Every frame retains some common support in these long runs. These distances condition on valid observations; omitted image regions and failed full-sequence methods are not assigned zero error. Point clouds are backprojected from predicted depth, intrinsics, and cameras, not from GT cameras or the predicted point head.

LoGeR inference timing. A dedicated H200 runs one unmeasured 32-frame warm-up and three fresh-state repetitions per sequence/length. CUDA synchronization brackets native segment forward execution (including internal input transfers) and output transfer to CPU; native boundary registration is also timed. Model loading, image decoding, checksum checks, focal fitting, disk writes, scoring, and inter-segment cleanup are excluded. Effective FPS is total unique frames across all repeats divided by their accumulated measured time. Other GPUs may run unrelated tasks, but no second model shares the measured GPU. Allocated and reserved peaks are CUDA allocator measurements, not persistent state size. Device-memory sampling every 0.5 s gives observed maxima, not exact device-wide peaks.

Table E1: LoGeR (Zhang et al., 2026) + SWA under the long KITTI-360 (Liao et al., 2023) 16/4-window, reset300 protocol. Times are mean ± population SD across three repeats; peaks are maxima. Allocated/reserved values are GiB, distinct from the decimal GB in the 32-frame resource table.

Frames	Sequence	Time (s)	FPS	Allocated	Reserved
1000	test_2	49.44 ± 0.8700	20.23	14.39	15.96
1000	test_3	49.05 ± 0.2900	20.39	14.39	15.96
2000	test_2	98.98 ± 0.3600	20.21	14.39	16.07
2000	test_3	98.90 ± 0.3400	20.22	14.39	16.07

These measurements characterize the reported LoGeR implementation and timing boundary, not an equal-resolution speed comparison with Mamba-VGGT.

E.1 VBR EVALUATION PROTOCOL

Sequences and evaluation. We evaluate Campus, DIAG, and Ciampino from VBR (Brizi et al., 2024). The public-GT `Campus_train0` and `DIAG_train0` routes contain 12,042 and 10,021 available observations, respectively; Ciampino contains 18,846 frames. Campus camera poses are obtained by interpolating public LiDAR poses and composing the provided rigid calibration. DIAG's camera matrices are checked against the same LiDAR-to-camera composition. Only undistorted left RGB images enter inference; GT poses and LiDAR depths do not condition predictions or window registration. ATE is translation RMSE after one whole-sequence Sim(3) evaluation alignment.

Campus and DIAG sampling. We select 10,000 Campus frames and 8000 DIAG frames spanning each complete valid route. Sampling uses adjacent-image motion only. Grayscale images are resized to width 160; median Farnebäck flow magnitude, excluding a four-pixel border, is clipped to $[0, 10]$.

Flow parameters are pyramid scale 0.5, three levels, window size 15, three iterations, polynomial size 5, and polynomial sigma 1.2. Interval weights are $0.5 + 0.5 f_i / \bar{f}$, with uniform weights when $\bar{f} \leq 10^{-6}$. Equally spaced cumulative-weight targets are interpolated to indices, rounded, and made strictly increasing while retaining both endpoints. All methods reuse the same saved inputs and evaluation ground truth.

Ciampino sampling-density comparison. Ciampino is evaluated with a 12,000-frame temporal subset and the full 18,846-frame sequence. Within each setting, all three methods receive identical inputs and use the same inference protocol as the main table. The full-sequence setting retains every available frame in temporal order, without temporal subsampling. Windowing and overlap do not remove input frames. These settings represent two input densities on one additional route, not two independent routes. Mamba-VGGT has lower ATE than both baselines in both settings, showing that its advantage is not confined to subsampled inputs.

Matched windows, resets, and stitching. All four settings in Table 5 use W16/O4 and reset300. Mamba-VGGT uses the same main checkpoint as the other evaluations (Appendix A.1), without VBR adaptation. Segments reset every 300 *new* observations and retain four preceding frames for boundary alignment; the first two intervals are $[0, 300)$ and $[296, 600)$. Mamba-VGGT clears all four states and restarts the leak5 counter; after five windows, taps 16/22 use retention 0.8. VGGT-W16 bypasses the adapters and restarts only its window grid. LoGeR resets TTT while preserving SWA history; internal `reset_every` is zero to avoid a second reset schedule. Leak5 applies only to Mamba-VGGT, not to the baselines.

Stitching uses predicted overlap cameras and rigid, fixed-scale transforms without GT. Methods retain native preprocessing, so resolution and compute are not matched. On Campus/DIAG, Mamba-VGGT/VGGT use 518×266 inputs and LoGeR uses 714×350 inputs (width$\times$height).

Additional LoGeR configurations. On the same Campus/DIAG inputs, LoGeR's reported VBR schedule (Zhang et al., 2026) uses W48/O3, resets TTT every five windows, retains SWA, and applies feedforward pose alignment at reset boundaries; reset300 is not additionally applied. ATEs are 64.95/33.62 m, versus 61.21/32.86 m for Mamba-VGGT under W16/O4 with reset300. For completeness, W32/O3 with reset300 and four-frame boundary overlap gives LoGeR 76.55/28.41 m on Campus/DIAG. Its DIAG result is better than Mamba-VGGT's W16/O4 result; Mamba-VGGT also using W32/O3 obtains 25.22 m with the unchanged checkpoint and leak5. These additional configurations are separate from the W16/O4 results in the main table.

End-to-end runtime and GPU memory. Resource measurements apply only to Campus with 10,000 frames, W16/O4, and reset300. Mamba-VGGT/VGGT-W16/LoGeR take 650.02/660.37/1086.76 s, including loading, image decoding, inference, merging, disk writes, and scoring. Peak PyTorch allocated/reserved memory is 6.398/7.887, 6.396/7.885, and 10.969/11.928 GiB, respectively. These are process-level allocator peaks, not device-wide peaks or persistent-state sizes. All methods process segments/windows and transfer outputs to CPU. Different resolutions and implementations prevent attributing resource gaps solely to state representation. One sequence length does not establish constant total GPU/CPU memory or disk storage; no repeat-run variance is reported. These measurements do not cover Ciampino or the additional window configurations.

Qualitative reconstruction. Figure 6 uses the same 10,000-frame Campus input and W16/O4 configuration as the main comparison. Clouds use predicted cameras and predicted depth/intrinsics for Mamba-VGGT or predicted local points for LoGeR. Each method's whole-input pose Sim(3) evaluation alignment is also applied to its cloud. A common normalized 24×48 image grid per frame is intersected over finite, positive-depth predictions. Up to 450,000 shared samples are displayed with a common view and scale, without ICP, error-based clipping, or window-wise GT corrections. Display subsampling does not shorten inference; these images do not establish quantitative reconstruction accuracy or complete scene coverage.