AdaVGGT: Self-Supervised Test-Time Adaptation for Video-based 3D Reconstruction

Anonymous authors
Paper under double-blind review

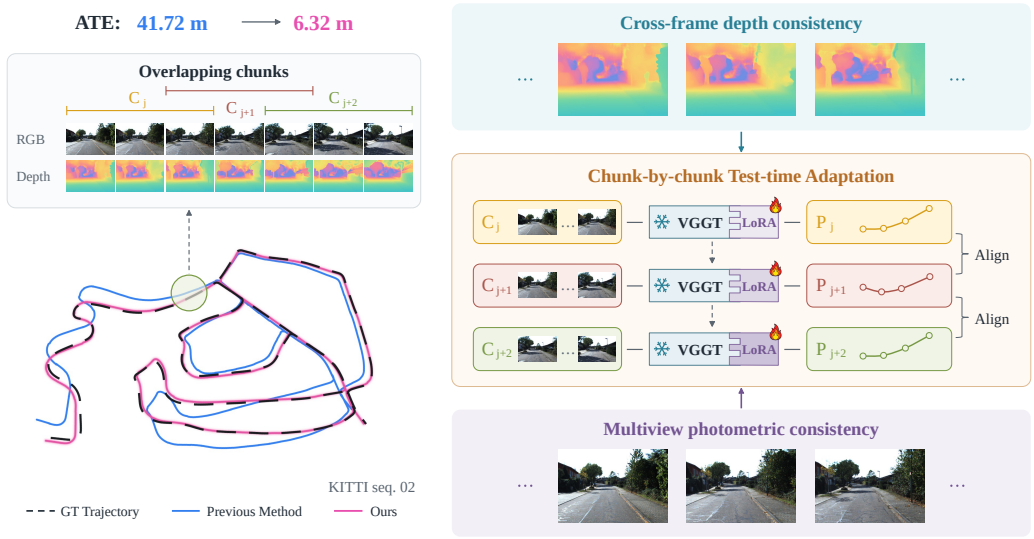

Figure 1: **Self-supervised online test-time adaptation for video 3D reconstruction.** Left: Estimated trajectories on KITTI Seq. 02 for raw VGGT (blue) and AdaVGGT (pink) against ground truth (dashed); ATE drops from 41.72 m to 6.32 m. The top panel shows overlapping temporal chunks ($\mathcal{C}_j, \mathcal{C}_{j+1}, \mathcal{C}_{j+2}$). Right: Low-rank adapters on depth and camera heads are optimized online via photometric and depth consistency, with adapter weights and optimizer states propagated across chunks.

Abstract

Feed-forward geometry models reconstruct 3D scenes in a single pass but repeatedly applying frozen weights to long videos leaves scene-specific errors uncorrected. Conversely, traditional per-scene optimization aligns observations to a target scene but lacks general priors and incurs heavy computation. We bridge these two paradigms with AdaVGGT, a self-supervised online test-time adaptation method that specializes pretrained predictors to a test video on the fly. We freeze the backbone of VGGT and jointly optimize lightweight low-rank (LoRA) corrections on its depth and camera pose heads during inference. Our objective requires no offline training, auxiliary datasets, or external optical flow; instead, it refines predictions directly on current observations via multi-view photometric reprojection and cross-frame depth consistency. To process long sequences online, adaptation proceeds causal-sequentially on overlapping temporal chunks, where each chunk inherits the optimized parameters and optimizer momentum from its predecessor. Across five real-world and synthetic outdoor, driving, and indoor video benchmarks, AdaVGGT consistently improves camera tracking and depth estimation over the unadapted foundation model. Notably, on KITTI Odometry, adaptation cuts the mean tracking error of VGGT from 18.09 m to 6.15 m—a 66% reduction—outperforming both classical vSLAM baselines and existing test-time optimization methods. These results demonstrate how strong general priors and scene-specific self-supervision can work together to enable robust, pixel-aligned 3D reconstruction at deployment. Code and models will be publicly released.

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.

1 INTRODUCTION

Reconstructing 3D scene and camera trajectories from video is a central problem in computer vision. Existing approaches generally follow two distinct paradigms: per-scene optimization and feed-forward prediction. Per-scene optimization methods (Mur-Artal et al., 2015; Teed & Deng, 2021) follow an analysis-by-synthesis strategy, iteratively refining scene pointclouds and camera poses until they explain the observations. While accurate and scene-adaptive, they require heavy computation and lack general priors. In contrast, recent feed-forward models (Wang et al., 2025a; 2024; 2025b) predict geometry in a single forward pass. They provide strong geometric priors and high-quality initial reconstructions. However, because their weights remain frozen at deployment, feed-forward models cannot adapt to specific scenes or resolve pixel-level geometric errors across frames.

We aim to combine the strengths of both paradigms. We want a framework that preserves the strong initialization of feed-forward models while retaining the ability of per-scene optimization to align observations to the target scene. The video itself provides a natural supervision signal: warping one frame into another view should reproduce the observed image, and depths from overlapping views should describe consistent 3D points.

Built upon VGGT (Wang et al., 2025a), we propose AdaVGGT, an online test-time adaptation method for video 3D reconstruction. We freeze VGGT's base weights and learn low-rank (LoRA) corrections (Hu et al., 2022) on its depth and camera heads directly during inference. Moreover, adaptation relies entirely on the current video observations without auxiliary training stages or offline adaptation datasets. Using calibrated camera intrinsics, we jointly optimize these corrections with *multi-view photometric reprojection* and *cross-frame depth consistency*. To process long videos online, adaptation operates on overlapping temporal chunks. Each chunk inherits the LoRA weights and Adam optimizer state from its predecessor, allowing learned corrections on earlier frames to assist later predictions. The entire process requires only the input video, without external optical flow or geometry annotations. Figures 1 and 2 illustrate the core idea and pipeline.

We evaluate AdaVGGT across five benchmarks spanning outdoor driving, synthetic environments, and indoor capture: KITTI Odometry, VKITTI2, KITTI Depth, ScanNet++, and 7Scenes. AdaVGGT consistently improves camera poses and median-scaled depth over raw VGGT. On KITTI Odometry, it cuts mean post-loop ATE from $18.09\,\mathrm{m}$ to $6.15\,\mathrm{m}$, a 66% error reduction, outperforming the strongest classical vSLAM baseline (ORB-SLAM3, $16.30\,\mathrm{m}$) and learning-based test-time adaptation methods like VGGT-Align ($19.99\,\mathrm{m}$). Ablations show that carrying both adapter parameters and optimizer state is critical for compounding gains along the video.

Our main contributions are twofold:

1. Formulation. We formulate online test-time adaptation for feed-forward geometry models. By learning low-rank head corrections on rolling video chunks, our method maintains strong pretrained priors while adapting to scene-specific observations.

2. Consistent gains across benchmarks. We demonstrate that self-supervised photometric and depth consistency signals yield substantial improvements in both camera tracking and depth estimation across five public datasets, exhibiting clear test-time scaling behavior.

2 RELATED WORK

Per-scene geometric optimization. Classical systems estimate scene geometry by fitting parameters to observed images, using direct photometric alignment (Engel et al., 2018) or sparse feature correspondences with reprojection error (Mur-Artal et al., 2015; Schönberger & Frahm, 2016). Learned approaches follow a similar structure: CodeSLAM (Bloesch et al., 2018) and BA-Net (Tang & Tan, 2019) optimize compact depth codes or basis coefficients alongside camera poses, whereas DROID-SLAM (Teed & Deng, 2021) and MegaSaM (Li et al., 2025) embed learned updates within dense bundle adjustment. In these methods, test-time optimization refines only scene-specific variables while keeping the network weights frozen. In contrast, we optimize corrections directly on the predictor, allowing geometric corrections learned on earlier frames to generalize to later ones. See *Sup.Mat.*'s Sec. D for extended discussion on direct SLAM and learned representation optimization.

Feed-forward geometry and long-video reconstruction. DUSt3R (Wang et al., 2024) and MASt3R (Leroy et al., 2024) regress dense pointmaps from image pairs. VGGT (Wang et al., 2025a) and π^3 (Wang et al., 2026) extend feed-forward prediction to multi-view sequences, jointly estimating depth and camera poses. To process long videos, VGGT-Long (Deng et al., 2026) aligns overlapping chunks using loop constraints, while CUT3R (Wang et al., 2025b) and StreamVGGT (Zhuo et al., 2026) track history through recurrent states or cached features. These methods provide strong geometric priors across scenes, yet their parameters remain fixed at deployment. AdaVGGT preserves this feed-forward initialization and adapts the predictor directly on the test video. *Sup.Mat.*'s Sec. D provides further details on streaming and memory-augmented reconstruction models.

Test-time learning of geometric context. Recent reconstruction methods update internal model representations at inference to retain scene context. TTT3R (Chen et al., 2026b) introduces confidence-guided updates to a recurrent state without extra training. Other approaches, including LoGeR (Zhang et al., 2026a), ZipMap (Jin et al., 2026), and Scal3R (Xie et al., 2026) train fast-weight memory architectures offline and update memory states at test time using feature association. These updates adjust how context is stored and retrieved rather than refining geometric prediction heads. In contrast, our updates directly adjust prediction heads using output-level photometric and depth-consistency residuals. We discuss test-time context architectures further in *Sup.Mat.*'s Sec. D.

Test-time adaptation of geometry predictors. Self-supervised photometric view synthesis and cross-frame depth consistency were originally developed for offline training of depth and ego-motion networks (Zhou et al., 2017; Godard et al., 2019; Bian et al., 2019). Recent works adapt predictors at test time using external optical flow (Luo et al., 2020; Zhang et al., 2022; Lee et al., 2022) or experience replay (Kuznietsov et al., 2021). For geometry foundation models, Free Geometry (Dai & Yang, 2026) optimizes multi-view attention with feature-level self-consistency, Self-Geometry (Youn et al., 2026) uses 2D point correspondences, and VGGT-Align (Zhang et al., 2026b) adapts normalization layers. Unlike prior work, AdaVGGT directly optimizes low-rank corrections on both depth and camera heads via calibrated photometric reprojection and internal depth consistency, carrying learned adapter weights and optimizer states across temporal chunks. *Sup.Mat.*'s Sec. D reviews additional work on online depth adaptation and foundation model fine-tuning.

3 Method

3.1 Overview

Given video frames $\{I_t\}_{t=1}^{T}$ and camera intrinsics $K \in \mathbb{R}^{3\times3}$, AdaVGGT estimates per-frame metric depth $D_t \in \mathbb{R}^{H\times W}$ and camera poses $E_t \in \mathrm{SE}(3)$ online. Built on VGGT (Wang et al., 2025a; Deng et al., 2026), we divide the sequence into overlapping temporal chunks $\{\mathcal{C}_j\}$ of size w and stride $w-o$, sharing o frames between adjacent chunks (e.g., $w=60, o=30$ yields frames 1–60, 31–90, and 61–120). For each chunk $\mathcal{C}_j$, the frozen backbone extracts visual features once. We then optimize low-rank (LoRA) corrections on the depth and camera heads for N_j steps using multi-view photometric reprojection and cross-frame depth consistency (Fig. 2). The adapted chunk outputs final depth maps and local camera poses, while its learned adapter parameters and optimizer state propagate to chunk $\mathcal{C}_{j+1}$ as a warm start. Sections 3.2 to 3.4 detail our head parameterization, self-supervised objectives, and sequential state propagation with trajectory composition.

3.2 Head Parameterization via Low-Rank Adaptation

We freeze all base weights of VGGT and inject low-rank (LoRA) adapter matrices (Hu et al., 2022) exclusively into the prediction heads. For a weight matrix $W_\ell^0 \in \mathbb{R}^{m_\ell\times n_\ell}$ at layer ℓ, the adapted weight is parameterized as:

$$W_\ell = W_\ell^0 + B_\ell A_\ell, \tag{1}$$

where $A_\ell \in \mathbb{R}^{r\times n_\ell}$ and $B_\ell \in \mathbb{R}^{m_\ell\times r}$ with low rank $r \ll \min(m_\ell, n_\ell)$.

We denote the trainable adapter parameters in the depth and camera pose heads as Φ_d and Φ_p, respectively, forming the full set of test-time parameters $\Phi = (\Phi_d, \Phi_p)$. Initializing all $B_\ell = 0$ guarantees that the network precisely reproduces its pretrained feed-forward predictions at the start of adaptation. Layer coverage, rank choices, and initialization details are provided in *Sup.Mat.*'s Sec. B.

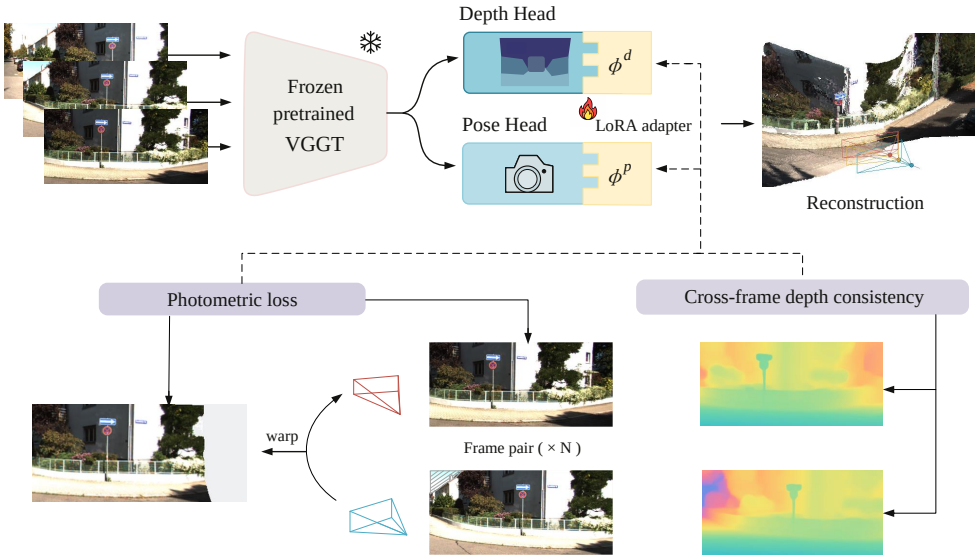

Figure 2: **Overview of AdaVGGT adaptation framework.** Given video frames and camera intrinsics, the frozen backbone extracts features once. Predicted depth and camera poses warp source views into target frames. Composite validity masks filter out unreliable regions, and the resulting photometric and depth-consistency residuals jointly update low-rank adapters on both prediction heads.

Because both depth and camera motion jointly govern the geometric warp between frames, we optimize Φ_d and Φ_p simultaneously. Since the heavy backbone features are computed once and cached, each test-time optimization step evaluates only the lightweight prediction heads.

3.3 Self-Supervised Geometric Supervision

Without ground truth, we adapt the network by enforcing cross-view consistency between adjacent frames within each chunk $\mathcal{C}_j$. We define a candidate set of directed frame pairs $\mathcal{P}_j = \{(t, s) \mid |t - s| \in \{1, 3, 5\}\}$, where I_t and I_s denote the target and source frames, respectively.

Let $T_{t\to s} = [R_{t\to s} \mid t_{t\to s}]$ with $R_{t\to s} \in \mathrm{SO}(3)$ and $t_{t\to s} \in \mathbb{R}^3$ be the relative pose from target frame t to source frame s. For a pixel $p \in I_t$ with homogeneous coordinates $\bar{p} = [u, v, 1]^\top$ and predicted depth $D_t(p)$, we warp p into the source view:

$$p' = \pi\left(K\left(R_{t\to s}\, D_t(p)\, K^{-1}\bar{p} + t_{t\to s}\right)\right), \tag{2}$$

where $\pi(\cdot)$ is perspective division that maps homogeneous pixel coordinates to the image plane. Let $D^{\mathrm{warp}}_{t\to s}(p)$ denote the depth of the warped 3D point in the source camera frame, i.e., the third coordinate of $R_{t\to s}D_t(p)K^{-1}\bar{p} + t_{t\to s}$. Based on this geometric warp, we use two complementary self-supervised residuals for each pixel p, where $\alpha \in [0, 1]$ weights the SSIM term against the ℓ_1 term and ε is a small constant for numerical stability:

$$\rho_{t,s}(p) = \frac{\alpha}{2}\Big(1 - \mathrm{SSIM}\big(I_t, I_{s\to t}\big)(p)\Big) + (1-\alpha)\big\|I_t(p) - I_{s\to t}(p)\big\|_1, \tag{3}$$

$$e_{t,s}(p) = \frac{\big|D^{\mathrm{warp}}_{t\to s}(p) - D_s(p')\big|}{D^{\mathrm{warp}}_{t\to s}(p) + D_s(p') + \varepsilon}. \tag{4}$$

Here, $I_{s\to t}$ denotes the source image warped into the target view via Eq. (2), with bilinear sampling throughout. $\mathrm{SSIM}(\cdot,\cdot)(p)$ evaluates the SSIM map over a local window centered at p, while the ℓ_1 term compares pixel-wise RGB colors. The photometric residual $\rho_{t,s}(p)$ provides joint gradient signals to both depth and camera heads. The cross-frame depth consistency residual $e_{t,s}(p)$ (Bian et al., 2019) penalizes discrepancies between the warped depth $D^{\mathrm{warp}}_{t\to s}(p)$ and the predicted source depth $D_s(p')$ to ensure scale coherence across overlapping views.

Robust Pixel Selection. To filter out occlusions, moving objects, and textureless skies, we compute a composite validity mask:

$$M_{t,s}(p) = M^{\mathrm{proj}}_{t,s}(p) \cdot M^{\mathrm{sky}}_{t,s}(p) \cdot M^{\mathrm{auto}}_{t,s}(p) \cdot M^{\mathrm{sc}}_{t,s}(p), \tag{5}$$



Figure 3: **Composite validity masking on a representative KITTI frame pair.** Target and source frames, warped source image, and filtered outlier regions: sky (gray), photometric auto-masking (orange), and scale-consistency outliers (pink).

where $M_{t,s}^{\text{proj}}(p)$ retains pixels that project within source image boundaries with $D_{t\to s}^{\text{warp}}(p) > 0$. $M_{t,s}^{\text{sky}}(p)$ excludes sky regions detected by SAM3 (Carion et al., 2026) in outdoor scenes (set to 1 indoors). The auto-masking $M_{t,s}^{\text{auto}}(p)$ and scale-consistency mask $M_{t,s}^{\text{sc}}(p)$ are defined as:

$$M_{t,s}^{\text{auto}}(p) = \mathbf{1}\left[\rho_{t,s}(p) < \rho_{t,s}^{\text{id}}(p)\right], \qquad M_{t,s}^{\text{sc}}(p) = \text{erode}(\mathbf{1}[e_{t,s}(p) < \tau]), \tag{6}$$

where $\mathbf{1}[\cdot]$ is the indicator function, $\text{erode}(\cdot)$ denotes morphological erosion, and τ is a depth-consistency threshold. Here $\rho_{t,s}^{\text{id}}(p)$ is the unwarped photometric error between I_t and I_s, filtering out static regions and occlusions (Godard et al., 2019), while erosion in $M_{t,s}^{\text{sc}}(p)$ removes uncertain object boundaries. Figure 3 illustrates the composite masking effect.

Joint Objective. Not all frame pairs are informative: some pairs share too little reliable content, e.g. when almost the entire view is occluded or masked out. Let Ω denote the full pixel grid of the target frame ($H \times W$ pixels). We admit a pair (t, s) into the active set $\mathcal{P}_j^+ \subseteq \mathcal{P}_j$ only if its valid pixel ratio exceeds a threshold γ:

$$\mathcal{P}_j^+ = \left\{(t,s) \in \mathcal{P}_j \;\middle|\; \frac{1}{|\Omega|}\sum_{p\in\Omega} M_{t,s}(p) \geq \gamma\right\}. \tag{7}$$

The chunk loss first averages the masked residuals within each admitted pair (so every pair contributes equally regardless of its valid area), then averages uniformly across pairs:

$$\mathcal{L}_j(\Phi) \triangleq \frac{1}{|\mathcal{P}_j^+|}\sum_{(t,s)\in\mathcal{P}_j^+} \frac{\sum_p M_{t,s}(p)\left[\rho_{t,s}(p;\Phi) + \lambda\, e_{t,s}(p;\Phi)\right]}{\sum_p M_{t,s}(p) + \varepsilon}, \tag{8}$$

where the residuals depend on Φ through the adapted predictions $\{D_t, E_t\}$, λ balances the two terms, and gradients are not backpropagated through mask computations.

3.4 Sequential State Propagation and Trajectory Composition

Sequential State Propagation. To enable continuous online learning along a video stream, chunk $\mathcal{C}_j$ inherits an adaptation state tuple $S_{j-1} = (\Phi_{j-1}, m_{j-1}, v_{j-1}, k_{j-1})$ from chunk $\mathcal{C}_{j-1}$, where Φ represents the LoRA parameters, and (m, v, k) denote the first moment, second moment, and step counter of the Adam optimizer (Kingma & Ba, 2015).

Starting from S_{j-1}, we perform N_j optimization steps on chunk $\mathcal{C}_j$ by minimizing $\mathcal{L}_j(\Phi)$. After updating, we run a single forward pass with the optimized parameters Φ_j to obtain the final depth maps and local camera poses for the chunk. The updated state $S_j = (\Phi_j, m_j, v_j, k_j)$ is then passed to chunk $\mathcal{C}_{j+1}$ Carrying both adapter parameters and optimizer momentum provides a warm start that amortizes adaptation across the video stream, as demonstrated in Sec. 4.5.

Global Trajectory Composition. Each chunk estimates depth and camera motion within its local coordinate system. For full video evaluation, we align adjacent chunks using the o overlapping frames. Using the predicted 3D pointmaps of the shared frames, we compute a $\text{Sim}(3)$ transformation $T_{j\to j-1} = (s_j, R_j, t_j)$ via robust registration (Zhang et al., 2026b). The cumulative transformation $G_j = G_{j-1} \circ T_{j\to j-1}$ maps local chunk predictions into the global coordinate frame of the first chunk. On KITTI Odometry, standard pose graph optimization with loop closure is applied to refine the global trajectory. For indoor benchmarks, chunk predictions are evaluated directly following standard protocols. Complete stitching and evaluation settings are detailed in *Sup.Mat.*'s Sec. B.

Table 1: **Camera tracking on KITTI Odometry.** ATE RMSE [m] (↓). `Req.` in Calib. denotes use of provided camera intrinsics. § marks published baseline numbers. LC denotes loop closure; Avg* excludes Seq. 01. Raw and adapted models use identical RANSAC chunk stitching. Baselines retain native settings. OOM denotes out of memory.

	Method	Calib.	Online	LC	Avg.	Avg.*	00	01	02	03	04	05	06	07	08	09	10
SLAM	ORB-SLAM2 (w/ LC)	Req.	✓	✓	54.82	9.46	6.03	508.34	14.76	1.02	1.57	4.04	11.16	2.19	38.85	8.39	6.63
	ORB-SLAM3	Req.	✓	✓	16.30	7.55	8.94	103.76	24.90	**0.76**	0.79	5.33	12.87	2.55	**4.45**	8.39	6.57
	LDSO	Req.	✓	✓	22.43	23.50	9.32	**11.68**	31.98	2.85	1.22	5.10	13.55	2.96	129.02	21.64	17.36
	DROID-SLAM	Req.	✓	–	100.28	75.85	92.10	344.60	107.61	2.38	1.00	118.50	62.47	21.78	161.60	72.32	118.70
	DPV-SLAM++	Req.	✓	✓	25.75	27.14	8.30	11.86	39.64	2.50	**0.78**	5.74	11.60	1.52	110.90	76.70	13.70
Geometry	CUT3R	No	✓	–	–	–	OOM	OOM	OOM	148.07	22.31	OOM	OOM	OOM	OOM	OOM	OOM
	VGGT	No	–	–	–	–	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM
	VGGT-Long (60)	No	–	✓	26.36	19.30	8.06	96.96	34.16	6.83	4.16	9.15	4.68	2.68	63.15	32.24	27.87
	VGGT-Long (90)	No	–	✓	22.72	20.94	11.97	40.51	49.85	5.41	2.86	9.88	6.07	3.47	66.27	32.27	21.33
	VGGT-Align§	No	–	✓	19.99	15.49	7.74	65.11	47.04	5.12	2.55	4.97	4.21	4.48	43.98	23.11	11.66
	VGGT (raw, zero adapter)	Req.	–	✓	18.09	12.99	7.55	69.09	41.72	5.83	1.08	4.68	5.49	2.22	32.86	17.92	10.61
Other	Scal3R	No	–	✓	14.55	–	–	–	–	–	–	–	–	–	–	–	–
	GPA-VGGT	Req.	–	–	–	–	–	–	–	–	–	–	–	12.54	–	21.43	–
TTA	TTT3R§	No	✓	–	177.73	–	–	–	–	–	–	–	–	–	–	–	–
	TTT3R (our protocol)	No	✓	✓	184.18	151.10	173.51	514.96	253.11	135.26	13.22	152.45	133.79	64.57	254.26	193.85	137.01
	VGG-T^3	Req.	–	✓	47.95	46.59	52.29	61.57	179.71	5.08	3.12	46.83	30.29	12.58	52.78	55.49	27.77
	AdaVGGT (ours)	Req.	✓	✓	**6.15**	**3.22**	**2.71**	35.49	**6.32**	1.80	0.92	**1.89**	**1.57**	**1.35**	6.09	**6.50**	**3.04**

4 Experiments

4.1 Experimental Setup

Datasets and Metrics. We evaluate AdaVGGT across five video reconstruction benchmarks covering diverse motion patterns and scene structures:

- **Outdoor Driving:** KITTI Odometry (Geiger et al., 2012) (11 real-world driving sequences) and KITTI Depth (Uhrig et al., 2017) (13 camera streams, 500 evaluation frames).
- **Synthetic Driving:** Virtual KITTI 2 (VKITTI2) (Cabon et al., 2020) (100 multi-view driving streams comprising 1,340 chunks).
- **Indoor Handheld Capture:** ScanNet++ (Yeshwanth et al., 2023) (20 dense indoor scenes, 386 chunks) and 7Scenes (Shotton et al., 2013) (7 handheld capture sequences, 39 chunks).

For outdoor sequences, we evaluate camera tracking via Absolute Trajectory Error (ATE RMSE [m]) before and after loop closure, and depth via median-scaled Absolute Relative error (AbsRel). For indoor sequences, we follow CommonBench (Zhang et al., 2026b) and evaluate pose accuracy using Area Under the Curve (AUC@τ for $\tau \in \{30°, 15°, 5°, 3°\}$) and median-scaled depth AbsRel. Complete metric definitions and aggregation protocols are described in *Sup.Mat.*'s Sec. B.

Baselines. We compare against classical vSLAM methods (ORB-SLAM2/3 (Mur-Artal & Tardós, 2017; Campos et al., 2021), LDSO (Gao et al., 2018), DROID-SLAM (Teed & Deng, 2021), DPV-SLAM++ (Lipson et al., 2024)), feed-forward multi-view models (CUT3R (Wang et al., 2025b), VGGT (Wang et al., 2025a), VGGT-Long (Deng et al., 2026)), and test-time learning/adaptation techniques (TTT3R (Chen et al., 2026b), VGG-T^3 (Elflein et al., 2026), Free Geometry (Dai & Yang, 2026), VGGT-Align (Zhang et al., 2026b), and Scal3R (Xie et al., 2026)). For rigorous comparison, the *VGGT (raw)* baseline evaluates the exact same chunked pipeline and stitcher as AdaVGGT with zero adapter corrections ($B_\ell = 0$), sharing identical calibration, data splits, and metric code.

4.2 Outdoor Driving Benchmarks

KITTI Odometry. Table 1 presents tracking performance across all eleven KITTI sequences. With $N_j = 70$ adaptation updates per chunk, AdaVGGT cuts mean post-loop ATE from 18.09 m to 6.15 m, a 65.99% error reduction over raw VGGT. Crucially, AdaVGGT improves tracking on all eleven individual sequences. Even before loop closure, adaptation lowers mean trajectory error from 31.78 m to 8.87 m, demonstrating that the primary gain stems from locally consistent geometry rather than subsequent pose-graph stitching. Furthermore, AdaVGGT outperforms all specialized vSLAM baselines (including ORB-SLAM3, 16.30 m) and feed-forward long-video approaches. Figure 4

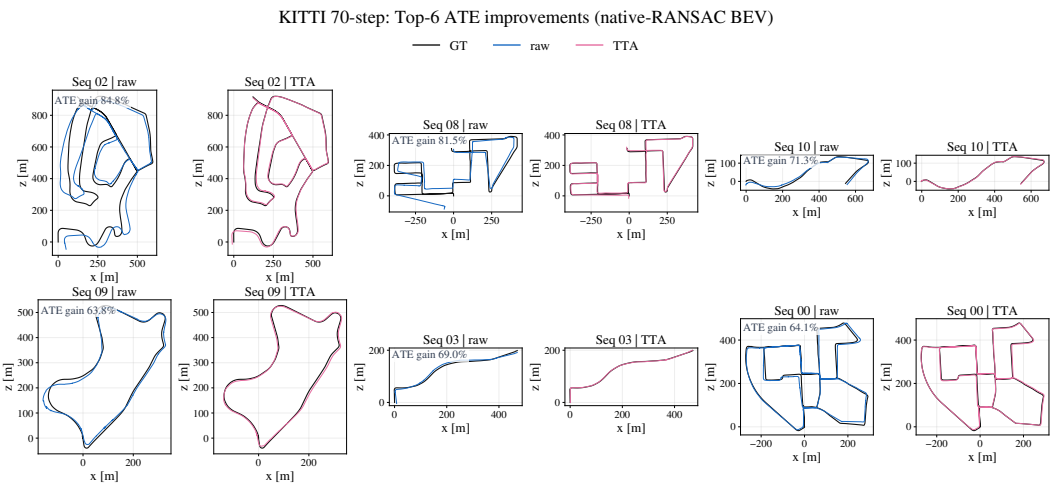

Figure 4: **Bird's-eye view trajectories on KITTI Odometry.** Reconstructed camera paths for sequences 02, 08, 10, 06, 03, and 00. Ground truth is shown in black, raw VGGT in blue, and AdaVGGT in pink. Both raw and adapted trajectories use identical RANSAC chunk stitching.

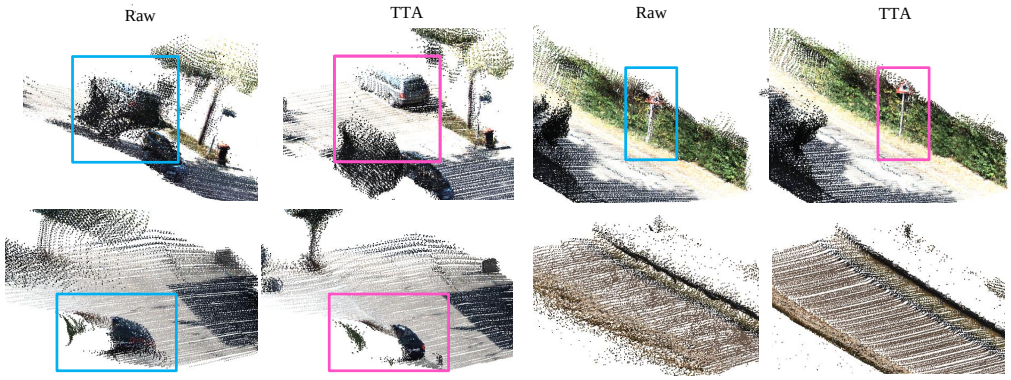

Figure 5: **Point-cloud reconstructions on KITTI Odometry.** 3D point clouds from raw VGGT (left) versus AdaVGGT (right) rendered under identical viewpoints and camera parameters.

Table 2: **KITTI Depth.** Median-scaled AbsRel.

Method	AbsRel ↓
VGGT (raw)	<u>0.064</u>
TTT3R	0.087
VGG-T^3	0.067
AdaVGGT (ours)	**0.060**

Table 3: **VKITTI2 tracking and depth.** Mean ATE [m] and median-scaled AbsRel.

Method	Global ATE (m) ↓	Local ATE (m) ↓	AbsRel ↓
VGGT (raw)	<u>1.19</u>	<u>0.12</u>	0.065
TTT3R	31.46	2.39	0.077
VGG-T^3	3.61	0.46	<u>0.061</u>
AdaVGGT (ours)	**1.11**	**0.07**	**0.055**

visualizes the bird's-eye-view trajectories of six representative sequences: the adapted prediction closely follows the ground-truth path where raw VGGT drifts substantially.

Beyond trajectories, Figure 5 compares point-cloud reconstructions of representative KITTI frames under identical viewpoints and rendering settings. After adaptation, many objects are visibly sharper and more consistent with their true structure, and the road surface is reconstructed more cleanly, demonstrating that the learned corrections improve not only camera trajectories but also the underlying scene geometry.

Table 4: **ScanNet++ evaluation.** Grey text indicates that offline training included the evaluation dataset.

Method	AUC@30 ↑	AUC@15 ↑	AUC@5 ↑	AUC@3 ↑	AbsRel ↓
VGGT (raw)	0.820	0.702	0.443	0.314	0.043
TTT3R	0.498	0.313	0.100	0.049	0.052
VGG-T^3	0.639	0.460	0.193	0.108	0.040
Free Geometry	**0.861**	**0.764**	**0.546**	**0.433**	**0.039**
AdaVGGT (ours)	0.843	0.741	0.520	0.408	0.041

Table 5: **7Scenes evaluation.** Pose AUC and median-scaled depth AbsRel.

Method	AUC@30 ↑	AUC@15 ↑	AUC@5 ↑	AUC@3 ↑	AbsRel ↓
VGGT (raw)	0.772	0.618	0.295	0.178	0.073
TTT3R	0.700	0.498	0.171	0.085	0.071
VGG-T^3	0.662	0.477	0.179	0.089	0.071
Free Geometry	0.786	0.638	**0.331**	**0.208**	0.071
AdaVGGT (ours)	**0.796**	**0.656**	0.320	0.188	**0.067**

Table 6: **Orthogonality with feature-matching adaptation.** Starting from Free Geometry predictions on ScanNet++ and 7Scenes, respectively, AdaVGGT runs 10 adaptation steps per chunk.

Dataset	Method	AUC@30 ↑	AUC@15 ↑	AUC@5 ↑	AUC@3 ↑	AbsRel ↓
ScanNet++	Free Geometry	0.861	0.764	0.546	0.433	0.039
	AdaVGGT + Free Geometry	**0.875**	**0.786**	**0.572**	**0.458**	**0.038**
7Scenes	Free Geometry	0.786	0.638	0.331	0.208	0.071
	AdaVGGT + Free Geometry	**0.804**	**0.668**	**0.354**	**0.220**	**0.069**

KITTI Depth. As shown in Tab. 2, AdaVGGT reduces median-scaled depth AbsRel from 0.064 to 0.060 across 13 camera streams. This confirms that jointly adapting depth and pose heads produces mutually beneficial geometric corrections rather than trading depth fidelity for camera accuracy. Representative RGB, ground-truth, raw, and adapted depth maps are shown in Fig. 6.

4.3 Synthetic Driving Video

On VKITTI2 (Tab. 3), AdaVGGT reduces global ATE from 1.19 m to 1.11 m and cuts local chunk ATE from 0.12 m to 0.07 m (a 40.2% relative improvement), while decreasing depth AbsRel from 0.065 to 0.055. The substantial gain in local chunk ATE underscores that test-time adaptation directly resolves chunk-level metric drift.

4.4 Indoor Handheld Capture

ScanNet++ and 7Scenes. Tables 4 and 5 report pose AUC and depth accuracy on complex indoor handheld videos. On ScanNet++, AdaVGGT improves pose AUC over raw VGGT across all thresholds, while reducing depth AbsRel. On 7Scenes, AdaVGGT yields consistent pose gains across all thresholds and achieves the lowest depth AbsRel (0.067) among all methods. Figure 6 presents qualitative depth comparisons: adapted depth maps recover sharper object boundaries and more consistent surfaces than raw VGGT across indoor and outdoor benchmarks.

Orthogonality to Feature-Matching Adaptation. To verify that our output-level geometric adaptation is orthogonal to feature-level adaptation, we initialize with Free Geometry's released aggregator LoRA weights and perform 10 additional update steps using our framework. As shown in Tab. 6, AdaVGGT + Free Geometry further improves AUC and reduces depth AbsRel on both datasets, demonstrating clear complementarity.

4.5 Ablation Studies and Analysis

We ablate the causal state carried across chunk boundaries on KITTI Odometry. Additional ablations on the LoRA adapter rank and validity masks are provided in *Sup.Mat.*'s Sec. C.

Causal State Propagation. Comparing four variants on all eleven sequences—*direct geometry* (optimizing per-frame depth and poses without adapting model weights), *reset* (reinitializing both adapters and Adam state per chunk), carrying only the adapter parameters Φ, and *full carry* (Φ plus optimizer state (m, v, k))—direct geometry barely improves over raw (18.09 $\rightarrow$ 17.24 m), confirming that updates should be made in the parameter space. Reset reaches 8.25 m, carrying Φ alone lowers error to 6.70 m, and full carry achieves the best 6.15 m (Fig. 7 (a)). This confirms that adapter parameters provide essential cross-chunk memory, while optimizer momentum ensures smooth optimization.

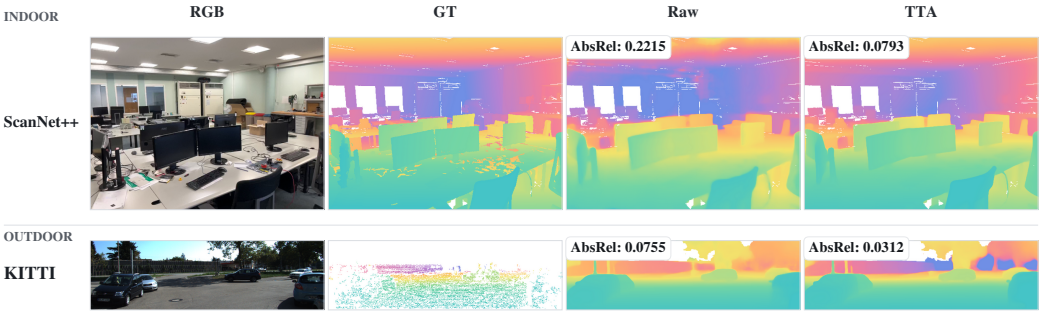

Figure 6: **Qualitative depth estimation results.** Visual comparisons across 7Scenes, ScanNet++, KITTI, and VKITTI2 showing input RGB, ground truth, raw VGGT, and AdaVGGT adapted depth maps.

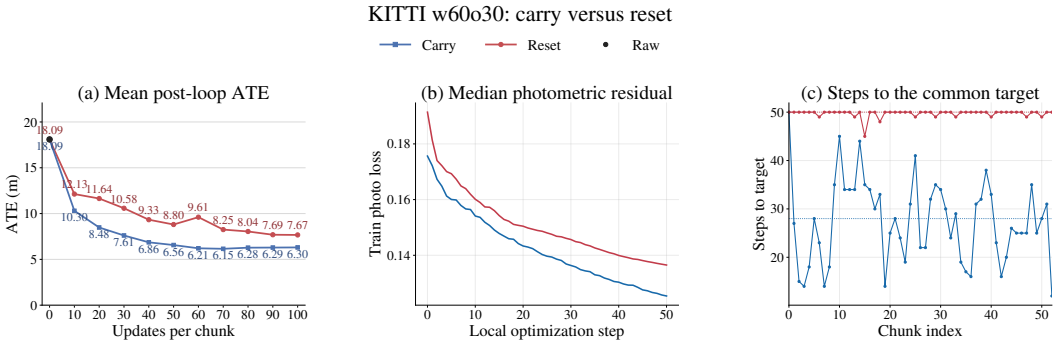

Figure 7: **Carry versus reset across vedio chunks**. (a) Mean post-loop ATE over KITTI Odometry sequence 00-10 as the number of updates per chunk increases; *carry* consistently outperforms *reset*. (b) Median photometric residual. (c) Number of updates required to reach per-chunk common residual target for seq09 under the $w60/o30$ setting with 50 updates per chunk. Blue and red denote *carry* and *reset* respectively.

Optimization Dynamics. Figure 7 (b)(c) traces the median photometric residual across update steps. The carried state initiates each chunk from a lower residual and reaches the common target threshold in substantially fewer iterations than reset, indicating that historical updates provide an effective warm start rather than causing temporal overfitting. This confirms positive forward transfer rather than chunk-local overfitting: the rolling updates capture persistent scene-level geometry and effectively turn historical observations into an informed warm start for subsequent online alignment.

Runtime and Efficiency Analysis. On KITTI ($w60/o30$) on an NVIDIA GPU, a raw VGGT forward pass requires $5.75\,\mathrm{s}$ per chunk. One step of lightweight adapter update takes only $0.42\,\mathrm{s}$ ($\sim 7\%$ of forward pass). Our method could easily balance cost and performance given different budget as shown in Figure.7.

5 Discussion and Conclusion

This paper presented AdaVGGT, a self-supervised online test-time adaptation method for video 3D reconstruction. Keeping the VGGT backbone frozen, AdaVGGT jointly adapts the depth and camera heads with lightweight low-rank corrections, supervised only by multi-view photometric reprojection and cross-frame depth consistency computed on the video itself. Adaptation proceeds causally over overlapping chunks, carrying both adapter parameters and optimizer momentum forward, so corrections learned on earlier frames benefit later predictions. Across five benchmarks, AdaVGGT consistently improves camera tracking and depth over raw VGGT—most notably reducing the mean post-loop ATE on KITTI Odometry by 66%, outperforming classical vSLAM and existing test-time adaptation methods.Our method currently relies on sliding split of a long video. This approach is still expensive when the video is long. We will explore streaming extension of our work in the future.

REFERENCES

Jiawang Bian, Zhichao Li, Naiyan Wang, Huangying Zhan, Chunhua Shen, Ming-Ming Cheng, and Ian Reid. Unsupervised scale-consistent depth and ego-motion learning from monocular video. In *Advances in Neural Information Processing Systems*, volume 32, 2019. 3, 4

Michael Bloesch, Jan Czarnowski, Ronald Clark, Stefan Leutenegger, and Andrew J. Davison. CodeSLAM—learning a compact, optimisable representation for dense visual SLAM. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2560–2568, 2018. 2, 16

Yohann Cabon, Naila Murray, and Martin Humenberger. Virtual KITTI 2. *arXiv preprint arXiv:2001.10773*, 2020. 6

Carlos Campos, Richard Elvira, Juan J. Gómez Rodríguez, José M. M. Montiel, and Juan D. Tardós. ORB-SLAM3: An accurate open-source library for visual, visual–inertial, and multimap SLAM. *IEEE Transactions on Robotics*, 37(6):1874–1890, 2021. doi: 10.1109/TRO.2021.3075644. 6

Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, Jie Lei, Tengyu Ma, Baishan Guo, Arpit Kalla, Markus Marks, Joseph Greer, Meng Wang, Peize Sun, Roman Rädle, Triantafyllos Afouras, Effrosyni Mavroudi, Katherine Xu, Tsung-Han Wu, Yu Zhou, Liliane Momeni, Rishi Hazra, Shuangrui Ding, Sagar Vaze, Francois Porcher, Feng Li, Siyuan Li, Aishwarya Kamath, Ho Kei Cheng, Piotr Dollár, Nikhila Ravi, Kate Saenko, Pengchuan Zhang, and Christoph Feichtenhofer. SAM 3: Segment anything with concepts. In *International Conference on Learning Representations*, 2026. 5

Lin-Zhuo Chen, Jian Gao, Shangzhan Zhang, Yihang Chen, Ka Leong Cheng, Yipengjing Sun, Liangxiao Hu, Nan Xue, Xing Zhu, Yujun Shen, Yao Yao, and Yinghao Xu. LingBot-Map: Geometric context transformer for streaming 3D reconstruction. *arXiv preprint arXiv:2604.14141*, 2026a. 17

Xingyu Chen, Yue Chen, Yuliang Xiu, Andreas Geiger, and Anpei Chen. TTT3R: 3D reconstruction as test-time training. In *International Conference on Learning Representations*, 2026b. 3, 6, 17

Zhuoguang Chen, Minghui Qin, Tianyuan Yuan, Zhe Liu, and Hang Zhao. LONG3R: Long sequence streaming 3D reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5273–5284, 2025. 16

Jan Czarnowski, Tristan Laidlow, Ronald Clark, and Andrew J. Davison. DeepFactors: Real-time probabilistic dense monocular SLAM. *IEEE Robotics and Automation Letters*, 5(2):721–728, 2020. doi: 10.1109/LRA.2020.2965415. 16

Yuhang Dai and Xingyi Yang. Free Geometry: Refining 3D reconstruction from longer versions of itself. *arXiv preprint arXiv:2604.14048*, 2026. 3, 6, 17

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. In *IEEE International Conference on Robotics and Automation*, 2026. 3, 6

Sven Elflein, Ruilong Li, Sérgio Agostinho, Zan Gojcic, Laura Leal-Taixé, Qunjie Zhou, and Aljosa Osep. VGG-T^3: Offline feed-forward 3D reconstruction at scale. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026. 6, 17

Jakob Engel, Thomas Schöps, and Daniel Cremers. LSD-SLAM: Large-scale direct monocular SLAM. In *European Conference on Computer Vision*, 2014. 16

Jakob Engel, Vladlen Koltun, and Daniel Cremers. Direct sparse odometry. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(3):611–625, 2018. doi: 10.1109/TPAMI.2017.2658577. 2, 16

Xiang Gao, Rui Wang, Nikolaus Demmel, and Daniel Cremers. LDSO: Direct sparse odometry with loop closure. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2018. 6

Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the KITTI vision benchmark suite. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2012. 6

Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel J. Brostow. Digging into self-supervised monocular depth estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3828–3838, 2019. 3, 5

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. 2, 3

Haian Jin, Rundi Wu, Tianyuan Zhang, Ruiqi Gao, Jonathan T. Barron, Noah Snavely, and Aleksander Hołyński. ZipMap: Linear-time stateful 3D reconstruction via test-time training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21748–21759, 2026. 3, 17

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. In *International Conference on 3D Vision*, 2026. 16

Kwonyoung Kim, Jungin Park, Jiyoung Lee, Dongbo Min, and Kwanghoon Sohn. PointFix: Learning to fix domain bias for robust online stereo adaptation. In *European Conference on Computer Vision*, pp. 568–585, 2022. doi: 10.1007/978-3-031-19839-7_33. 17

Diederik P. Kingma and Jimmy Lei Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015. 5

Yevhen Kuznietsov, Marc Proesmans, and Luc Van Gool. CoMoDA: Continuous monocular depth adaptation using past experiences. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 2907–2917, 2021. 3, 17

Yushi Lan, Yihang Luo, Fangzhou Hong, Shangchen Zhou, Honghua Chen, Zhaoyang Lyu, Bo Dai, Shuai Yang, Chen Change Loy, and Xingang Pan. STream3R: Scalable sequential 3D reconstruction with causal transformer. In *International Conference on Learning Representations*, 2026. 17

Yao-Chih Lee, Kuan-Wei Tseng, Guan-Sheng Chen, and Chu-Song Chen. Globally consistent video depth and pose estimation with efficient test-time training. *arXiv preprint arXiv:2208.02709*, 2022. 3

Vincent Leroy, Yohann Cabon, and Jerome Revaud. Grounding image matching in 3D with MASt3R. In *European Conference on Computer Vision*, 2024. 3, 16

Zhengqi Li, Richard Tucker, Forrester Cole, Qianqian Wang, Linyi Jin, Vickie Ye, Angjoo Kanazawa, Aleksander Holynski, and Noah Snavely. MegaSaM: Accurate, fast and robust structure and motion from casual dynamic videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10486–10496, 2025. 2, 16

Zizun Li, Jianjun Zhou, Yifan Wang, Haoyu Guo, Wenzheng Chang, Yang Zhou, Haoyi Zhu, Junyi Chen, Chunhua Shen, and Tong He. WinT3R: Window-based streaming reconstruction with camera token pool. In *International Conference on Learning Representations*, 2026. 17

Lahav Lipson, Zachary Teed, and Jia Deng. Deep patch visual SLAM. In *European Conference on Computer Vision*, 2024. 6

Changkun Liu, Jiezhi Yang, Zeman Li, Yuan Deng, Jiancong Guo, and Luca Ballan. Mem3R: Streaming 3D reconstruction with hybrid memory via test-time training. *arXiv preprint arXiv:2604.07279*, 2026. 17

Xuan Luo, Jia-Bin Huang, Richard Szeliski, Kevin Matzen, and Johannes Kopf. Consistent video depth estimation. *ACM Transactions on Graphics*, 39(4), 2020. 3

Raúl Mur-Artal and Juan D. Tardós. ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras. *IEEE Transactions on Robotics*, 33(5):1255–1262, 2017. doi: 10.1109/TRO.2017.2705103. 6

Raúl Mur-Artal, J. M. M. Montiel, and Juan D. Tardós. ORB-SLAM: A versatile and accurate monocular SLAM system. *IEEE Transactions on Robotics*, 31(5):1147–1163, 2015. doi: 10.1109/TRO.2015.2463671. 2, 16

Richard A. Newcombe, Steven J. Lovegrove, and Andrew J. Davison. DTAM: Dense tracking and mapping in real-time. In *Proceedings of the IEEE International Conference on Computer Vision*, 2011. 16

Johannes L. Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4104–4113, 2016. 2, 16

You Shen, Zhipeng Zhang, Yansong Qu, Xiawu Zheng, Jiayi Ji, Shengchuan Zhang, and Liujuan Cao. FastVGGT: Fast visual geometry transformer. In *International Conference on Learning Representations*, 2026. 17

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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2930–2937, 2013. 6

Chengzhou Tang and Ping Tan. BA-Net: Dense bundle adjustment networks. In *International Conference on Learning Representations*, 2019. 2, 16

Zachary Teed and Jia Deng. DROID-SLAM: Deep visual SLAM for monocular, stereo, and RGB-D cameras. In *Advances in Neural Information Processing Systems*, volume 34, 2021. 2, 6, 16

Alessio Tonioni, Oscar Rahnama, Thomas Joy, Luigi Di Stefano, Thalaiyasingam Ajanthan, and Philip H. S. Torr. Learning to adapt for stereo. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9661–9670, 2019. 17

Jonas Uhrig, Nick Schneider, Lukas Schneider, Uwe Franke, Thomas Brox, and Andreas Geiger. Sparsity invariant CNNs. In *International Conference on 3D Vision*, 2017. 6

Hengyi Wang and Lourdes Agapito. 3D reconstruction with spatial memory. In *International Conference on 3D Vision*, pp. 78–89, 2025. doi: 10.1109/3DV66043.2025.00013. 16

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. 2, 3, 6, 16

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. 2, 3, 6, 17

Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. DUSt3R: Geometric 3D vision made easy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20697–20709, 2024. 2, 3, 16

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. In *International Conference on Learning Representations*, 2026. 3, 16

Yuqi Wu, Wenzhao Zheng, Jie Zhou, and Jiwen Lu. Point3R: Streaming 3D reconstruction with explicit spatial pointer memory. In *Advances in Neural Information Processing Systems*, volume 38, 2025. 16

Tao Xie, Peishan Yang, Yudong Jin, Yingfeng Cai, Wei Yin, Weiqiang Ren, Qian Zhang, Wei Hua, Sida Peng, Xiaoyang Guo, and Xiaowei Zhou. Scal3R: Scalable test-time training for large-scale 3D reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21760–21771, 2026. 3, 6, 17

Congrong Xu, Huachen Gao, Xingyu Chen, Yuliang Xiu, Jun Gao, and Anpei Chen. R^3: 3D reconstruction via relative regression. *arXiv preprint arXiv:2605.26519*, 2026a. 17

Yangfan Xu, Lilian Zhang, Xiaofeng He, Pengdong Wu, Wenqi Wu, and Jun Mao. GPA-VGGT: Adapting VGGT to large scale localization by self-supervised learning with geometry and physics aware loss. *arXiv preprint arXiv:2601.16885*, 2026b. 17

Jianing Yang, Alexander Sax, Kevin J. Liang, Mikael Henaff, 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*, pp. 21924–21935, 2025. 16

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. 6

Seokhyun Youn, Dahyeon Kye, Sung-Ho Bae, and Jihyong Oh. Self-Geometry: GT-free and plug-and-play test-time adaptation for geometrically consistent 3D vision foundation models. *arXiv preprint arXiv:2608.10708*, 2026. 3, 17

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. 17

Yuheng Yuan, Qiuhong Shen, Shizun Wang, Xingyi Yang, and Xinchao Wang. Test3R: Learning to reconstruct 3D at test time. In *Advances in Neural Information Processing Systems*, volume 38, 2025. 17

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*, 2026a. 3, 17

Wei Zhang, Yihang Wu, Songhua Li, and Qi Wang. VGGT-Align: Bridging local reconstruction and global consistency for long-sequence 3D reconstruction. *arXiv preprint arXiv:2608.15260*, 2026b. 3, 5, 6, 14, 17

Zhenyu Zhang, Stéphane Lathuilière, Andrea Pilzer, Nicu Sebe, Elisa Ricci, and Jian Yang. Online adaptation through meta-learning for stereo depth estimation. *arXiv preprint arXiv:1904.08462*, 2019. 17

Zhoutong Zhang, Forrester Cole, Zhengqi Li, Michael Rubinstein, Noah Snavely, and William T. Freeman. Structure and motion from casual videos. In *European Conference on Computer Vision*, pp. 20–37, 2022. doi: 10.1007/978-3-031-19827-4_2. 3

Lei Zhou, Haoyu Wu, Akshat Dave, and Dimitris Samaras. Learning 3D reconstruction with priors in test time. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026. 17

Tinghui Zhou, Matthew Brown, Noah Snavely, and David G. Lowe. Unsupervised learning of depth and ego-motion from video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 3

Dong Zhuo, Wenzhao Zheng, Jiahe Guo, Yuqi Wu, Jie Zhou, and Jiwen Lu. Streaming visual geometry transformer. In *International Conference on Learning Representations*, 2026. 3, 17

Appendix

A Use of Large Language Models 14

B Implementation and Training Details 14

C Additional Ablation Studies 15

D Detailed Related Work 16

E Mathematical Notation 17

A Use of Large Language Models

In this work, we used generative AI tools (GPT-5.6 Sol, Claude Opus 5.5, and Gemini 3.8 Flash) for three tasks with required disclosure: implementing parts of the method and evaluation code, supporting the qualitative analysis of reconstructions and trajectories, and helping interpret experimental results. We did not use generative AI tools for any other task with required disclosure under the ICLR 2027 policy. Additionally, we used these tools to draft parts of the paper, edit software code, and improve the readability of the manuscript.

We reviewed all AI-assisted work. All AI-assisted code was reviewed by the authors before use in the reported experiments. All reported numbers were taken directly from the outputs of our evaluation scripts. AI-suggested interpretations of results and qualitative observations were retained only after the authors checked them against the corresponding tables, figures, and ablations. The authors take full responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

B Implementation and Training Details

Evaluation Subsets and Data Preparation.

- **KITTI Odometry:** Sequences 00–10 are evaluated with ground-truth poses from GPS/INS.
- **VKITTI2:** 100 photorealistic video streams across diverse weather and lighting conditions, totaling 1,340 temporal chunks.
- **KITTI Depth:** 500 owned test frames from 13 camera streams evaluated against LiDAR ground truth.
- **ScanNet++:** 20 representative handheld sequences totaling 386 chunks.
- **7Scenes:** 7 indoor handheld sequences comprising 39 chunks.

Pose Metrics and Aggregation Protocols. For indoor datasets (ScanNet++ and 7Scenes), CommonBench (Zhang et al., 2026b) aligns each chunk to its first camera frame and evaluates all unordered frame pairs. Pairwise error is computed as the maximum of rotation error and translation angular error. AUC@τ measures the mean cumulative recall over one-degree histogram bins from 0° to τ° for $\tau \in \{30^\circ, 15^\circ, 5^\circ, 3^\circ\}$. Under indoor $w30/o15$, each non-final chunk contributes its first 15 frames and the final chunk remaining all frames, so every frame is evaluated once. We compute CommonBench AUC from all unordered pairs within each owner chunk, pool pair errors across chunks of each scene while excluding cross-chunk pairs because their local gauges differ, and average the resulting scene AUCs equally across frames.

For outdoor sequences (KITTI and VKITTI2), ATE RMSE measures the root mean square error of camera positions after $\mathrm{Sim}(3)$ trajectory alignment. Global ATE is evaluated on the fully composed sequence trajectory, while local chunk ATE represents the mean chunk-level error before trajectory composition.

Table 7: **Adaptation hyperparameters across benchmarks.** N_j denotes test-time update steps per chunk; LR is the learning rate shared by depth and pose heads.

Dataset	Window/overlap	N_j	LR	Frame stride	Δ	M^{auto}	Sky
KITTI Odometry	$w60/o30$	70	3×10^{-4}	1	$1, 3, 5$	per chunk	SAM3
VKITTI2	$w60/o30$	50	3×10^{-5}	1	$1, 3, 5$	per chunk	SAM3
KITTI Depth	$w10/o5$	50	3×10^{-4}	1	1	per chunk	SAM3
ScanNet++	$w30/o15$	50	10^{-4}	1	$1, 3, 5$	per update	–
7Scenes	$w30/o15$	50	10^{-4}	11	$1, 3, 5$	per update	–

Window Tails and Frame Ownership. A w/o video stream initiates chunks at indices $0, w - o, 2(w - o), \ldots$ until reaching the video end. The final chunk ingests all remaining frames without zero-padding. Chunk-level indoor evaluations include all overlap frames within each evaluated window, whereas KITTI Depth assigns each frame strictly to its first-owner chunk.

Shared Adaptation Hyperparameters. Across all settings, low-rank adapters of rank $r = 8$ are inserted into all Linear, Conv2d, and ConvTranspose2d layers of the depth and pose prediction heads. Depth and camera heads are jointly updated using functional Adam with learning rate η and momentum parameters $(\beta_1, \beta_2) = (0.9, 0.999)$. Adaptation losses are computed over directed frame pairs with temporal offsets $\Delta = \{1, 3, 5\}$ ($\Delta = \{1\}$ for KITTI Depth). All A_ℓ matrices are initialized from a fixed random normal distribution, while B_ℓ is initialized to zero to guarantee that the unadapted state strictly matches the pretrained base model.

Detailed Objective Formulations. Predicted depth maps are clamped to $[0.05\,\bar{D}_t, \min(10\,\bar{D}_t, 200\,\text{m})]$, where $\bar{D}_t = \text{median}(D_t)$. SSIM employs a 3×3 window, and L_1 is averaged across RGB channels with weighting $\alpha = 0.85$. We use $\lambda = 0.5$. The projection validity mask M^{proj} requires all pixels in a 3×3 neighborhood to project within the image boundaries with positive depth. The depth consistency threshold is set to $\tau = 0.1$ with a morphological erosion radius of 3. Denominators include $\varepsilon = 10^{-7}$ for numerical stability.

Per-Dataset Configurations. Table 7 summarizes the per-dataset hyperparameter configurations.

C Additional Ablation Studies

We conduct a comprehensive ablation study on KITTI Odometry Seq. 09, evaluating adapter rank, support-mask policies, and intrinsics under the standard configuration ($w60/o30$, 53 chunks and 1,591 frames, 70 updates per chunk, bidirectional frame pairs with $\Delta = \{1, 3, 5\}$, $\lambda = 0.5$, learning rate 3×10^{-4}, minimum pair support $\gamma = 0.15$, and RANSAC stitcher with 500 iterations at seed 0). All 11 runs completed all chunks with one loop constraint per chunk. In the following tables, *raw* denotes the zero-adapter output and *TTA* the adapted output; *pre*/*post* denote ATE before and after loop closure, both measured after chunk stitching. The raw output is regenerated per configuration, so small differences across raw rows are retained.

LoRA Rank Ablation. Table 8 reports camera tracking error across adapter ranks from $r = 1$ to $r = 64$. Every tested rank improves the raw trajectory after loop closure: TTA ATE ranges from 6.43 to 6.92 m versus raw 17.92 m. The dependence on rank is non-monotonic—rank 2 performs best on this sequence (6.43 m) while the main-text setting $r = 8$ is only 0.01 m higher (6.44 m). Increasing rank therefore does not consistently improve accuracy.

Mask and Intrinsics Ablations. Table 9 compares individual masking policies against the full composite validity mask, and examines the impact of estimated versus calibrated camera intrinsics. The *only* policies retain the named optional mask (in addition to projection validity) and disable the other two optional masks. The combined mask achieves the lowest post-loop ATE (6.44 m) versus automask-only (6.69 m), sky-only (6.93 m), and dynamic-SC-only (7.19 m). Disabling calibrated intrinsics causes large tracking drift (21.52 m), verifying the importance of grounded geometric calibration.

Table 8: **LoRA rank ablation on KITTI Odometry Seq. 09.** ATE RMSE in meters. All variants use composite validity masks and calibrated camera intrinsics.

Rank	Raw pre	TTA pre	Raw post	TTA post
$r = 1$	29.19	6.77	17.92	6.85
$r = 2$	29.19	8.67	17.92	**6.43**
$r = 4$	29.19	9.84	17.92	6.55
$r = 8$ (main)	29.19	7.48	17.92	6.44
$r = 16$	29.19	6.92	17.92	6.92
$r = 32$	29.19	9.44	17.92	6.83
$r = 64$	29.19	6.84	17.92	6.72

Table 9: **Validity mask and intrinsic ablation on KITTI Seq. 09.** Evaluated with rank $r = 8$. The *only* variants disable all optional masks except the specified one.

Variant	Raw pre	TTA pre	Raw post	TTA post
All masks, calibrated K (reference)	29.19	7.48	17.92	**6.44**
Sky only	29.19	6.95	17.92	6.93
Dynamic SC only	29.19	10.88	17.92	7.19
Automask only	29.19	5.22	17.92	6.69
All masks, predicted K	39.27	59.11	21.53	21.52

D Detailed Related Work

Photometric Consistency and Scene-Level Geometric Optimization. Image consistency has long supported joint reasoning about scene structure and camera motion. Direct methods such as DTAM, LSD-SLAM, and DSO estimate geometry through photometric alignment, with different tracking, mapping, and optimization schemes (Newcombe et al., 2011; Engel et al., 2014; 2018). Feature-based systems such as ORB-SLAM and COLMAP instead establish image correspondences and refine camera poses and scene points through geometric reprojection constraints (Mur-Artal et al., 2015; Schönberger & Frahm, 2016). In both cases, the variables adapted to a sequence describe its geometry.

Learned representations and optimizers build on this connection. CodeSLAM and DeepFactors optimize compact depth codes together with camera poses (Bloesch et al., 2018; Czarnowski et al., 2020), while BA-Net performs feature-metric bundle adjustment over poses and depth-basis coefficients (Tang & Tan, 2019). DROID-SLAM uses learned correspondence updates within dense bundle adjustment, and MegaSaM extends joint camera and disparity estimation to challenging casual videos (Teed & Deng, 2021; Li et al., 2025). These methods use learned components to guide geometric inference; their inference-time optimization updates codes or explicit geometry variables rather than fine-tuning the pretrained predictor on the target video.

Feed-Forward Geometry Models and Memory Architectures. DUSt3R (Wang et al., 2024) predicts dense pointmaps from image pairs and combines them through global alignment, while MASt3R (Leroy et al., 2024) augments this representation with dense local features for image matching. Fast3R (Yang et al., 2025) jointly reconstructs multiple views in a shared coordinate system in one forward pass. VGGT (Wang et al., 2025a) predicts camera parameters, depth, pointmaps, and tracks using alternating frame-wise and global attention. Complementary designs address reference-frame symmetry in π^3 (Wang et al., 2026) and optional geometric inputs in MapAnything (Keetha et al., 2026). These models provide strong geometric predictions from visual context. Building on VGGT, we use self-supervision from the observed video to jointly adjust its depth and pose predictors.

Long-video reconstruction models incorporate history into the predictor: Spann3R (Wang & Agapito, 2025) reads from spatial memory, Point3R (Wu et al., 2025) maintains explicit spatial pointers, and LONG3R (Chen et al., 2025) combines gated memory retrieval with coarse-to-fine

decoding. CUT3R (Wang et al., 2025b) updates a persistent latent state as observations arrive. StreamVGGT (Zhuo et al., 2026) and STream3R (Lan et al., 2026) use causal attention with cached historical features, while WinT3R (Li et al., 2026) combines sliding-window reconstruction with a historical camera-token pool. LingBot-Map (Chen et al., 2026a) separates anchor context, recent dense features, and compressed trajectory history. R^3 (Xu et al., 2026a) estimates relative poses against selected reference frames and fuses them using predicted confidence. FastVGGT (Shen et al., 2026) reduces the tokens processed by global attention, while InfiniteVGGT (Yuan et al., 2026) manages a rolling KV memory for causal streaming.

Test-Time Learning and Online Adaptation. Recent reconstruction methods use test-time updates to retain scene context in compact states. TTT3R (Chen et al., 2026b) derives confidence-guided updates of the recurrent state without additional training, while Mem3R (Liu et al., 2026) combines an implicit fast-weight memory for camera tracking with an explicit token memory for geometric mapping. LoGeR (Zhang et al., 2026a) combines fast weights with sliding-window attention for chunk-causal reconstruction. VGG-T^3 (Elflein et al., 2026) and ZipMap (Jin et al., 2026) fit feature associations in compact neural scene representations, and Scal3R (Xie et al., 2026) shares context across chunks through synchronized memory updates.

In stereo and monocular depth estimation, L2A (Tonioni et al., 2019) learns an initialization and confidence weighting for unsupervised stereo adaptation; OMLA (Zhang et al., 2019) combines online feature alignment with meta-learning; and PointFix (Kim et al., 2022) uses an auxiliary network during meta-training to improve initialization. For monocular video, CoMoDA (Kuznietsov et al., 2021) continually adapts depth and ego-motion networks while replaying past experience. For geometry foundation models, Test3R (Yuan et al., 2025) optimizes visual prompts using agreement between reconstructions of overlapping image pairs. Free Geometry (Dai & Yang, 2026) adapts multi-view transformer LoRA parameters and camera tokens by matching partial-view features to a frozen full-view branch. Self-Geometry (Youn et al., 2026) uses test-scene feature matches to impose multi-view and epipolar constraints through attention LoRA. TCO (Zhou et al., 2026) adapts the shared decoder with LoRA while keeping prediction heads fixed. VGGT-Align (Zhang et al., 2026b) complements geometric scale alignment with normalization-parameter adaptation under photometric, geometric, and temporal objectives. GPA-VGGT (Xu et al., 2026b) performs self-supervised post-training on unlabeled video using photometric and cross-view geometric constraints.

E Mathematical Notation

Table 10 lists key mathematical notations and symbols used throughout the paper.

Table 10: **Summary of mathematical notations.** Primary symbols and definitions used throughout the paper.

Symbol	Meaning
W_ℓ^0, W_ℓ	Base frozen weight and adapted weight at layer ℓ
A_ℓ, B_ℓ, r	LoRA adapter decomposition matrices and adapter rank ($r \ll \min(m_\ell, n_\ell)$)
Φ_d, Φ_p, Φ	Trainable adapter parameters for depth head, pose head, and full model ($\Phi = (\Phi_d, \Phi_p)$)
$\mathcal{C}_j, w, o$	j-th temporal video chunk; chunk window size; inter-chunk overlap frames
N_j, η	Number of test-time optimization updates per chunk; adaptation learning rate
I_t, D_t, E_t, K	RGB frame; predicted metric depth; world-to-camera pose $\in \mathrm{SE}(3)$; calibrated intrinsics
$\mathcal{P}_j, \mathcal{P}_j^+$	Candidate frame pairs and admitted frame pairs within chunk $\mathcal{C}_j$
$\rho_{t,s}, e_{t,s}$	Multi-view photometric residual and cross-frame depth consistency residual
$M_{t,s}$	Composite validity mask ($M^{\mathrm{proj}} \cdot M^{\mathrm{sky}} \cdot M^{\mathrm{auto}} \cdot M^{\mathrm{sc}}$)
m, v, k	Adam first-moment, second-moment buffers, and update step counter
S_j	Causal adaptation state tuple (Φ_j, m_j, v_j, k_j) propagated from chunk j to $j+1$
$T_{j\to j-1}, G_j$	Inter-chunk $\mathrm{Sim}(3)$ transformation and cumulative global trajectory alignment map