Rethinking the State Update Gate for Long-Sequence Recurrent 3D Reconstruction

Kejun Ren[1*] **Lei Jin**[1*] **Tianxin Huang**[2] **Lianming Xu**[1] **Li Wang**[1†]

[1]Beijing University of Posts and Telecommunications, Beijing, China
[2]School of Computing and Data Science, The University of Hong Kong

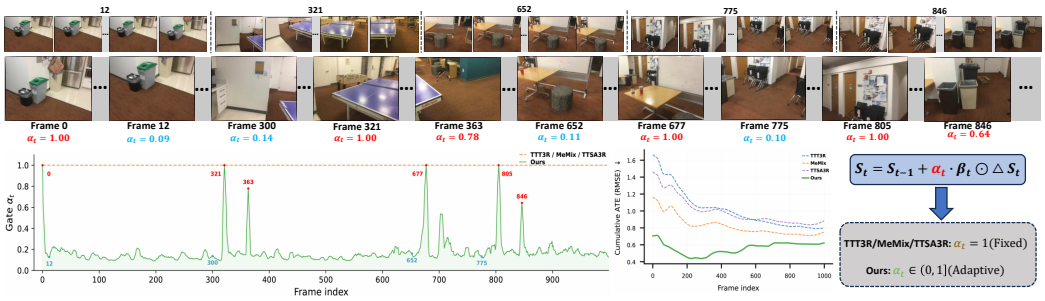


Figure 1: **Adaptive frame gating for streaming 3D reconstruction.** Existing methods (TTT3R, MeMix, TTSA3R) update the recurrent state S_t with $\alpha_t \equiv 1$ regardless of frame content, accumulating drift on long sequences. We introduce an adaptive frame gate $\alpha_t \in (0, 1]$ scaled by frame-to-frame feature change, requiring no parameters, training, or extra forward pass. The gate shown is AFG-Pose (Sec. 4.2). **Top:** Representative frames from a 1000+-frame indoor sequence. Red keyframes (e.g., #321, #805) at significant scene or viewpoint transitions receive large α_t ($\to 1$); cyan near-duplicates (e.g., #12, #300) receive small α_t (~ 0.1) and are strongly down-weighted (but never discarded). **Bottom-left:** Gate α_t over the sequence—low during stable motion, spiking at keyframes; existing methods stay at $\alpha_t \equiv 1$. **Bottom-middle:** Cumulative ATE—our method maintains a significantly lower and more stable error profile on long sequences, indicating effective drift suppression. **Bottom-right:** Our update rule (Eq. 4): α_t multiplies TTT3R's per-token gate β_t at the state-update site.

Abstract

Streaming 3D reconstruction under a strict constant-memory budget hinges on how the recurrent state is updated as the stream evolves. We profile TTT3R-style per-token gates across five benchmarks and discover a structural bottleneck: the gate is intrinsically bounded in magnitude (median 0.31; never exceeding 0.6) and nearly frame-invariant, yielding an effective memory horizon of only $\sim$3 frames per state token, which serves as the structural origin of long-sequence drift. We trace this to a missing axis: existing inference-time methods modulate updates only at the per-token, intra-frame level, while the orthogonal frame-level question of *how strongly each frame should contribute to the state* has been treated as content-independent. We close this gap with a scalar frame-level gate $\alpha_t \in (0, 1]$ derived in closed form from frame-to-frame changes of internal features—a graded write weight, inspired by classical Simultaneous Localization and Mapping (SLAM) keyframe selection, that never discards a frame and requires no parameters, no training, and no extra forward pass. Across six benchmarks spanning camera pose, video depth, and 3D reconstruction at sequence lengths up to $4,661$ frames, our gate cuts ATE by 51% on long TUM-RGBD pose sequences, reduces AbsRel by 13.0% on Bonn video

*Equal contribution. †Corresponding author.

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

depth, and on KITTI long-sequence pose estimation surpasses both LongStream and Keyframe-VO in average ATE, while retaining strictly constant memory at zero training cost. Code is available at `https://github.com/rkj12345/AFG`.

1 Introduction

Streaming 3D reconstruction processes an RGB video incrementally and produces camera poses and pixel-aligned geometry on the fly, under strict memory and latency constraints that grow with the input length [22, 31]. Offline batch methods such as DUSt3R [23], MASt3R [11], VGGT [21], and Fast3R [27] produce strong reconstructions through global attention over all frames, but their quadratic cost and the requirement that all frames be loaded simultaneously preclude online use on long sequences. Among streaming approaches, CUT3R [22] achieves a strictly constant memory footprint by compressing the full sequence history into a fixed-size recurrent state of N tokens, $S \in \mathbb{R}^{N \times d}$, that each incoming frame reads from and writes back into. KV-cache variants [31, 24] retain richer history but grow with sequence length, while sliding-window attention [12] keeps memory constant only by discarding early frames, making CUT3R the unique combination of constant memory and full-history retention. **Under a strict constant-memory budget, the central question becomes how this state should be updated as the stream evolves** (Fig. 1).

Recent inference-time methods all modulate this update along the same axis: a per-token gate $\beta_t \in [0,1]^N$ that decides which state slots absorb the current frame's residual. TTT3R [3] derives β_t from cross-attention logits as a constrained test-time gradient step; TTSA3R [30] reshapes it via attention-style gating; the concurrent MeMix [8] adds a hard token mask. Through systematic profiling on long sequences across five benchmarks, we discover a striking regularity: β_t is structurally bounded (median 0.31; no value exceeds 0.6), nearly invariant across datasets, and—most importantly—its variation *across* frames is less than half of its variation *within* a frame. Under an EMA reading of the update, this corresponds to an effective memory horizon of only $\sim$3 frames, structurally explaining the long-sequence drift these methods continue to exhibit. **Yet a more fundamental observation lies beneath this bound: all of these designs lack any frame-level adaptivity. They modulate updates only along the intra-frame axis, i.e., per-token within a single frame, and treat every frame as equally informative regardless of its content.**

Classical SLAM has, for two decades, treated content-dependent admission as a first-class problem under the name *keyframe selection* [2]: only frames bringing sufficient novelty enter the map, while redundant ones are skipped. Modern streaming reconstruction has quietly abandoned this idea, every I_t writes into the state with the same nominal strength regardless of content. We argue that this is the missing axis, and propose to reintroduce it not as a binary admit/skip decision but as a *graded write weight*: every frame still contributes, but with a strength that grows with its novelty. We borrow the signal of keyframe selection (how new a frame looks) rather than its role: keyframe selection bounds computation by choosing which frames enter a growing map, whereas our state has a fixed size and the problem is how strongly each frame overwrites it (Sec. 4.3). Concretely, we introduce a scalar frame-level gate $\alpha_t \in (0,1]$ that multiplicatively modulates the state-update residual. Computed in closed form from the frame-to-frame change of internal features the base model already produces, α_t adds no parameters and no extra forward pass. Most frames are redundant and receive small α_t that barely perturbs the state; only significant scene or viewpoint transitions drive α_t toward 1, where the update proceeds at full TTT3R strength.

The design of Adaptive Frame Gating (AFG) is minimal: a single multiplicative scalar at the state-update site. As a plug-in modifier, AFG preserves the constant-memory inference property of CUT3R-style backbones, unlike retraining-based alternatives such as LongStream [5]. Across six benchmarks (TUM-RGBD, ScanNet, Bonn, KITTI, 7-Scenes, and NRGBD) with sequences from 50 to 4661 frames, AFG improves over inference-time gating baselines including TTT3R, MeMix, and TTSA3R on long sequences; on long TUM-RGBD trajectories ($L \geq 600$) it reduces ATE by 51% over TTT3R, and on KITTI it surpasses both LongStream and the RL-trained Keyframe-VO [7].

Our contributions are summarized as follows.

- **Analysis of per-token gating.** We identify a key limitation of TTT3R-style per-token gates: they are nearly frame-invariant and imply an effective memory horizon of only $\sim$3 frames, suggesting that the lack of frame-level adaptivity is a key bottleneck in long-sequence streaming reconstruction.

- **Adaptive Frame Gating.** We propose AFG, a scalar frame-level gate $\alpha_t \in (0, 1]$ computed in closed form from features already produced by the base model. We instantiate it as AFG-Img and AFG-Pose, using encoder global features and decoder pose tokens, respectively. Both variants are parameter-free, training-free, require no extra forward pass, and preserve constant memory.
- **Extensive long-sequence evaluation.** We validate AFG across pose, depth, and reconstruction tasks on six benchmarks. On long sequences it improves over inference-time gating baselines and compares favorably with long-sequence methods based on retraining or learned keyframe policies.

2 Related Work

Feed-forward 3D reconstruction. Feed-forward methods bypass the conventional Structure-from-Motion (SfM) pipeline by directly regressing scene geometry from input images. DUSt3R [23] and MASt3R [11] cast pairwise reconstruction as dense pointmap regression in a shared coordinate frame; VGGT [21] and Fast3R [27] extend this to the multi-view setting through global Transformer attention. While these methods achieve high reconstruction quality, they require all input frames to reside in attention simultaneously, scaling quadratically with sequence length and precluding online use on long videos.

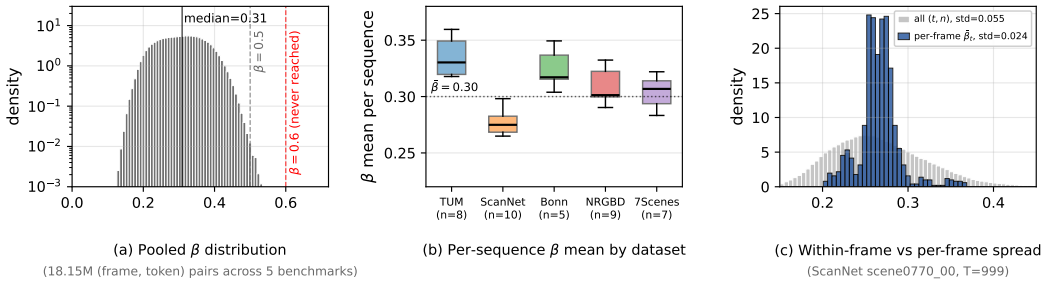
Streaming 3D reconstruction. To process inputs of arbitrary length, streaming methods turn to sequential or recurrent paradigms. Spann3R [20] introduces an explicit spatial memory for incremental reconstruction; CUT3R [22] formulates the problem as an RNN with a fixed-size recurrent state—the only design among streaming variants that strictly preserves constant memory. Subsequent works explore richer memory structures, including KV caches [31], point-level memory [24], and sliding-window attention [12], but their memory grows with sequence length and limits deployment on long videos. Within the constant-memory regime, how the recurrent state should be updated as the stream evolves remains an open question, since unfiltered residual accumulation has been shown to cause severe long-sequence drift [3].

Long-sequence adaptation strategies. Three complementary lines of work address this long-sequence degradation. The first relies on *architecture-level retraining*: a growing body of work retrains new architectures or memory mechanisms tailored for long-sequence streaming reconstruction [5, 4, 29, 10, 25, 13, 26]. As a representative, LongStream [5], built on the VGGT family, redesigns camera pose parameterization and KV-cache training to mitigate first-frame anchor bias and cache saturation. These approaches share the cost of giving up the constant-memory property and training from scratch. The second uses *external frame selection policies*: Keyframe-VO [7] freezes the base model and trains a reinforcement-learning policy that makes a discrete admit/skip decision for each frame, extending classical SLAM keyframe selection [14, 2] to feed-forward 3D models. The third modulates state updates at *inference time*: TTT3R [3] reinterprets the recurrent update as a constrained test-time gradient step [19] and derives per-token learning rates $\beta_t \in [0, 1]^N$ from cross-attention logits; TTSA3R [30] reshapes β_t via attention-style gating; the concurrent MeMix [8] adds a per-token hard mask. Inference-time methods preserve constant memory and require no training, but all operate exclusively on the per-token, intra-frame axis. **Our work belongs to this third line and adds the orthogonal frame-level axis**: compared to TTT3R, TTSA3R, and MeMix, we go beyond per-token modulation by making each frame's contribution to the state content-dependent at the frame level; compared to Keyframe-VO, we share the goal of content-dependent admission but replace its discrete RL-trained admit/skip policy with a graded write weight computed in closed form from the base model's internal features—requiring no training, no parameters, and no extra forward pass.

3 Structural Diagnosis of Per-Token Gating

3.1 Recurrent State Updates for Continuous 3D Reconstruction

Given a continuous image stream $\{I_t\}_{t=1}^{T}$, online 3D reconstruction estimates per-frame camera pose T_t, intrinsics K_t, and pixel-aligned pointmap P_t. CUT3R [22] casts this as a recurrent sequence model with a fixed-length state $S \in \mathbb{R}^{N \times d}$ initialized from learnable embeddings: each frame I_t



(a) Pooled β distribution
(18.15M (frame, token) pairs across 5 benchmarks)

(b) Per-sequence β mean by dataset

(c) Within-frame vs per-frame spread
(ScanNet scene0770_00, T=999)

Figure 2: **Empirical properties of TTT3R's per-token learning rate β. (a)** Density of β pooled over all (t, n) pairs from 39 sequences across 5 benchmarks (18.15M points). The distribution is sharply concentrated; *no* measurement exceeds $\beta = 0.6$. **(b)** Per-sequence mean of β grouped by dataset; despite the diverse motion regimes, all five benchmarks lie within $[0.27, 0.34]$. **(c)** For a representative long sequence (ScanNet `scene0770_00`, $T = 999$), the histogram of all (t, n) values (gray, $\mathrm{std} = 0.055$) is more than twice as wide as the histogram of per-frame means $\bar{\beta}_t$ (blue, $\mathrm{std} = 0.024$): TTT3R exhibits per-token variation, but the per-frame mean is essentially constant.

is tokenized into image tokens X_t, the state is updated as $S_t = \mathrm{Update}(S_{t-1}, X_t)$, and a read-out $Y_t = \mathrm{Read}(S_t, X_t)$ is decoded into (T_t, K_t, P_t) by a prediction head.

State–input interaction. Update and Read are jointly realized by an L-layer dual-stream cross-attention decoder. The state stream attends to image tokens at each layer:

$$S^{(l)} = S^{(l-1)} + \mathrm{softmax}\left(Q_S^{(l)} K_X^{(l)\top}\right) V_X^{(l)}, \tag{1}$$

with a symmetric stream updating $X^{(l)}$ from $S^{(l-1)}$. Stacking L layers yields the per-frame update $S_t = S_{t-1} + \Delta S_t$, where ΔS_t aggregates the cross-attention contributions of the current frame.

Continuous state update (CUT3R). CUT3R writes this residual back unconditionally: $S_t = S_{t-1} + \Delta S_t$. On long sequences with redundant content, this monotonic accumulation overwrites earlier observations and drives the state toward catastrophic drift [3].

Test-time learning for state update (TTT3R). TTT3R [3] reinterprets ΔS_t as a gradient step in a test-time training framework [19], with S acting as model parameters, X_t as a test sample, and ΔS_t as a descent direction on a self-supervised reconstruction loss. To regulate the step per state token, TTT3R averages the cross-attention logits over L decoder layers, H heads per layer, and K image tokens per frame, and applies the resulting per-token learning rate multiplicatively before integration:

$$\beta_t \;=\; \sigma\left(\frac{1}{LHK}\sum_{l,h,k} Q_{S_{t-1}}^{(l,h)} K_{X_t}^{(l,h,k)\top}\right) \;\in\; [0,1]^N, \qquad S_t = S_{t-1} + \beta_t \odot \Delta S_t. \tag{2}$$

We write $\beta_t^{(n)}$ for the gate value at frame $t \in \{1, \ldots, T\}$ on the n-th state token, $n \in \{1, \ldots, N\}$. Each state token now retains history in proportion to its current cross-attention relevance, mitigating the drift of CUT3R. Whether β_t provides effective control on long sequences—an empirical question not examined in prior work—is the focus of this section.

3.2 Empirical Properties of β: Bounded Magnitude and Negligible Frame Variation

We profile β_t across 5 benchmarks (TUM-RGBD, ScanNet, Bonn, NRGBD, 7-Scenes) covering indoor scanning, handheld, and dynamic scenes. Across all sequences and frames, we collect $\beta_t^{(n)}$ at every (t, n) pair and analyze its distribution along two axes: pooled magnitude (Fig. 2a, b) and within-frame versus per-frame variation (Fig. 2c).

β is structurally bounded. On the pooled distribution (Fig. 2a), β has median 0.31, 99% of its mass below 0.44, and a maximum of 0.558; *none* of the 18.15M measurements exceeds 0.6. This bound holds uniformly across datasets (Fig. 2b): per-sequence means lie in a tight band of $[0.27, 0.34]$, with no benchmark significantly higher or lower than the global mean of 0.31. The bound is therefore a property of the architecture rather than of any particular dataset.

β has negligible frame-to-frame variation. Within a single sequence, the spread of β across all (t, n) pairs is more than twice as large as the spread of the per-frame means $\bar{\beta}_t = \frac{1}{N}\sum_n \beta_t^{(n)}$ (Fig. 2c): on ScanNet `scene0770_00`, the within-frame std is 0.055 while the cross-frame std of $\bar{\beta}_t$ is only 0.024. The same pattern holds across the other benchmarks: the per-frame mean of β remains essentially constant regardless of sequence length or motion regime. While the per-token gate yields meaningful variation across the N state tokens (consistent with TTT3R's design intent), the per-frame mean is essentially constant: TTT3R provides token-level selectivity but *no* frame-level adaptivity.

Why β behaves this way. Both properties stem from the multi-axis averaging in Eq. 2: each $\beta_t^{(n)}$ is the sigmoid of an average over $L \times H \times K$ ($\approx$ 110K in the CUT3R-512 architecture) cross-attention logits. This averaging has two effects: (i) extreme logit values are diluted into a narrow effective range, and (ii) the summation over K image tokens fuses frame-specific spatial information into a global statistic, with the residual mean dominated by the pretrained (Q, K) projections rather than by X_t. Let $\bar{A}$ denote the averaged cross-attention logit inside the sigmoid of Eq. 2, i.e., $\bar{A} = \frac{1}{LHK}\sum_{l,h,k} Q_{S_{t-1}}^{(l,h)} {K_{X_t}^{(l,h,k)}}^\top$. Empirically, $\bar{A} \approx -0.83$ across all measurements, yielding $\sigma(\bar{A}) \approx 0.30$ and matching the three observations of Fig. 2. The consequence is that every frame contributes a similar moderate fraction ($\sim$30%) of its residual ΔS_t to the state regardless of content. We next examine what this implies for memory over long sequences.

3.3 Memory Horizon Implied by Bounded β

EMA decay and memory horizon. Writing the residual as $\Delta S_t = \tilde{S}_t - S_{t-1}$, where $\tilde{S}_t$ is the state the current frame would write on its own, the update of Eq. 2 is a convex combination, $S_t = (1 - \beta_t) \odot S_{t-1} + \beta_t \odot \tilde{S}_t$, i.e., exactly an exponential moving average on each state slot with smoothing factor $\beta_t^{(n)}$. The only assumption we add is that β is approximately constant in time, which is what Sec. 3.2 measures. The contribution of frame $t - k$ to $S_t^{(n)}$ then decays geometrically as $(1 - \beta)^k$, giving an effective memory horizon of approximately $1/\beta$ frames[1]. Since the per-frame mean is essentially constant across datasets ($\bar{\beta}_t \in [0.27, 0.34]$, Sec. 3.2):

$$\text{horizon} \;\approx\; 1/\bar{\beta} \;\approx\; 3 \text{ frames}. \tag{3}$$

This horizon is not only short but *stable*: it stays within $[2.9, 3.7]$ regardless of sequence length or motion regime. Over a sequence of $T = 1000$ frames, the contribution of frame 0 to S_{100} has already decayed to $(0.69)^{100} \approx 10^{-16}$—early observations are erased almost immediately and never recovered.

The horizon is an exact property of the update rule; the step from a short horizon to pose drift is not derived but established empirically, by the benchmarks of Sec. 5 and the controlled probe of Appendix A1. We argue that this short horizon is the structural cause of long-sequence drift: since β is content-independent (Sec. 3.2), every frame overwrites the state with equal strength, so an informative keyframe is displaced within $\sim$3 steps by whatever follows—including near-redundant frames. The memory is not just short, but *indiscriminately* short: early geometric evidence is forgotten long before it ceases to be relevant. We verify this mechanism via a controlled redundancy experiment in Appendix A1.

4 Adaptive Frame Gating

Building on the diagnosis of Sec. 3, we restore the missing frame-level adaptivity with a scalar gate α_t that adaptively scales β_t based on observation novelty.

[1] Defined as the lag at which the contribution drops to $1/e$ of its instantaneous value: $(1 - \beta)^k = 1/e \Rightarrow k \approx 1/\beta$ for small β.

4.1 Factorized State Update

We replace the TTT3R update of Eq. 2 with the factorized form

$$S_t \;=\; S_{t-1} \;+\; \alpha_t \cdot \beta_t \odot \Delta S_t, \tag{4}$$

where $\alpha_t \in (0, 1]$ is a scalar frame-level gate that adaptively scales the current frame's contribution to the state based on its content, and $\beta_t \in [0, 1]^N$ is TTT3R's per-token gate from Eq. 2 that determines *which* state tokens should receive the contribution. The two gates operate on orthogonal axes, token (spatial) and frame (temporal), and combine multiplicatively: $\beta_t^{(n)}$ controls per-token spatial selection while α_t modulates the per-frame update strength. This composition preserves TTT3R's token selectivity while adding an independent temporal modulation.

4.2 Frame Gate from CUT3R-Internal Features

We compute α_t from the frame-to-frame change of a feature that CUT3R already produces, yielding two natural instances. Both add zero parameters and zero forward passes; throughout, $\sigma(\cdot)$ denotes the logistic sigmoid $\sigma(z) = 1/(1 + e^{-z})$.

AFG-Img. We use the encoder's per-frame global feature $g_t \in \mathbb{R}^d$, defined as the spatial mean of the encoder's patch tokens, $g_t = \frac{1}{K}\sum_{k=1}^{K} X_t^{(k)}$, which CUT3R already employs as its LocalMemory query. Its frame-to-frame change drives:

$$\alpha_t^{\text{img}} \;=\; \sigma\big(\kappa_{\text{img}}\big(\|g_t - g_{t-1}\|_2 - \tau_{\text{img}}\big)\big)\,. \tag{5}$$

AFG-Pose. We use the first row of the final decoder-layer state, $p_t = S_{t,1}^{(L)} \in \mathbb{R}^d$, which feeds CUT3R's pose prediction head and captures the model's per-frame pose representation. Its frame-to-frame change drives:

$$\alpha_t^{\text{pose}} \;=\; \sigma\big(\kappa_{\text{pose}}\big(\|p_t - p_{t-1}\|_2 - \tau_{\text{pose}}\big)\big)\,. \tag{6}$$

Here τ is the centre of the sigmoid (a threshold on the feature change) and κ its sharpness; the two constants differ between variants because g_t and p_t live on different scales. We use $(\kappa_{\text{img}}, \tau_{\text{img}}) = (1, 1.5)$ and $(\kappa_{\text{pose}}, \tau_{\text{pose}}) = (3, 1.0)$ on every benchmark, with no per-scene tuning; the sensitivity to both constants is reported in Appendix A2. When the feature change is zero (a frame identical to its predecessor), α_t saturates at the strict positive lower bound $\sigma(-\kappa\tau) > 0$—$\sigma(-1.5) \approx 0.18$ for AFG-Img and $\sigma(-3) \approx 0.047$ for AFG-Pose—rather than collapsing to zero, so the gate remains in $(0, 1]$ on arbitrarily long redundant runs. The gate is capped at 1 by construction, so AFG can only weaken a write, never enlarge it beyond TTT3R's; we test this design choice in Appendix A3. The two variants capture novelty at different abstraction levels: g_t reflects raw visual content, p_t reflects the model's pose-related cognition. Empirically, AFG-Img excels on smooth-scanning regimes where pose changes are gentle, while AFG-Pose excels on dynamic regimes where pose changes carry the strongest signal (Sec. 5). We use them as complementary regime-specific variants, since naive fusion does not exceed the better single gate (Sec. 5.5).

Interpretation: a longer, content-aware horizon. α_t acts as an *inference-time learning rate* on the recurrent state: $\alpha_t \to 1$ on novel frames (absorbed at full TTT3R strength), $\alpha_t \to \alpha_{\min} = \sigma(-\kappa\tau)$ on redundant ones. This reshapes the memory horizon of Sec. 3: a keyframe is overwritten not at the fixed rate β, but at $\alpha_t\beta$ set by the novelty of *subsequent* frames. When followed by low-information frames ($\alpha_t \to \alpha_{\min}$), its contribution decays as $(1 - \alpha_{\min}\beta)^k$, extending the horizon from $1/\beta \approx 3$ frames to $1/(\alpha_{\min}\beta)$. Writing $H(\alpha) \approx 1/(\alpha\bar{\beta})$, TTT3R ($\alpha \equiv 1$) has $H \approx 3$ frames whatever the content; with the observed $\alpha_{\min} \approx 0.048$ on fully redundant inputs, H reaches $\sim$60 frames—up to a $\sim$20$\times$ extension under sustained redundancy. AFG thus does not remember harder; it *forgets selectively*, spending a limited memory budget on informative frames rather than the most recent ones. We verify this with a controlled redundancy-injection probe in Appendix A1, where the drift decay measured independently gives $\sim$64 frames.

4.3 Relation to Keyframe Selection

AFG shares its signal with classical keyframe selection but differs in role and mechanism. Keyframe selection exists because a map and its optimisation grow with the number of frames, so a subset must

be chosen and a rejected frame never enters the map. In a fixed-size recurrent state nothing needs to be excluded; the problem is that new frames keep overwriting old ones, so AFG sets *how strongly* each frame writes, and every frame writes. Two consequences follow. First, since $\alpha_t \geq \alpha_{\min} > 0$, a frame whose novelty is under-read is delayed rather than lost, which is why an adjacent-frame change suffices as the signal, whereas a binary rule must measure novelty against the map to avoid irrecoverable mistakes. Second, the difference is measurable (Appendix A4): binarising our gate with non-keyframes set to 0, i.e., the classical admit/skip rule, raises TUM-RGBD ATE from 0.052 to 0.061 and RPE_r from 0.647 to 0.880 relative to keeping them at a small weight, and a hard response is worse than our graded one on ScanNet (0.352 vs. 0.281 ATE for AFG-Pose). Measuring novelty against the last accepted frame instead of the previous frame, closer to classical keyframe criteria, is also worse (0.065 vs. 0.054 ATE; 0.092 vs. 0.086 AbsRel).

5 Experiments

Tasks and benchmarks. We evaluate on three streaming 3D reconstruction tasks: camera pose estimation on TUM-RGBD [18], ScanNet [6], and KITTI [9]; video depth estimation on Bonn [15]; and 3D reconstruction on 7-Scenes [17] and NRGBD [1]. Sequence lengths span 50 to 4661 frames. We adopt standard alignment conventions throughout: Sim(3) Umeyama alignment for camera pose, and metric alignment for depth (scale-and-shift depth results are additionally reported in Appendix A11.2 as a robustness check). Per-task evaluation details are given in each subsection.

Baselines. We compare against three families: inference-time gating methods (TTT3R [3], TTSA3R [30], MeMix [8]); feedforward backbones (CUT3R [22], StreamVGGT [31], FastVGGT [16], InfiniteVGGT [28]); and long-sequence-specialized methods (LongStream [5], Keyframe-VO [7]).

Implementation details. AFG is applied on top of TTT3R to the same frozen CUT3R checkpoint without fine-tuning, with the gate constants of Sec. 4.2 used unchanged across all benchmarks. All experiments are conducted on a single NVIDIA A100 (80 GB).

5.1 Camera Pose Estimation

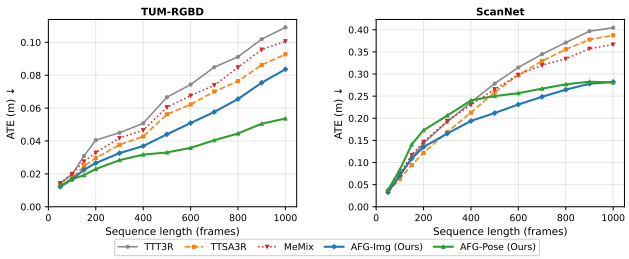


Figure 3: **Camera pose ATE vs. sequence length.** On long sequences AFG outperforms TTT3R, TTSA3R, and MeMix on TUM-RGBD and ScanNet, with the gap widening as length grows; on short ScanNet sequences TTSA3R is best.

Following prior work [22, 3, 8], we evaluate long-sequence absolute trajectory error (ATE) on TUM-RGBD and ScanNet as sequence length increases; results are shown in Fig. 3, with per-length numerical values tabulated in Appendix A8. On TUM-RGBD both variants outperform TTT3R, TTSA3R, and MeMix for $L \geq 200$; on ScanNet AFG-Img is best for $L \geq 300$, while AFG-Pose trails TTT3R up to $L = 400$ and overtakes all baselines from $L = 500$. The gap widens as sequences grow longer, and on short ScanNet sequences ($L \leq 200$), where little drift has accumulated, TTSA3R is best; MeMix offers limited additional gain over TTT3R throughout, supporting the analysis of Sec. 3 that token-level gating alone cannot address frame-level drift. AFG-Pose is the stronger variant on TUM-RGBD (dynamic walking sequences), reducing average ATE at $L \geq 600$ by 51% over TTT3R, while AFG-Img leads on ScanNet (smooth indoor scanning) with a 29% reduction.

5.2 Long-Sequence Pose Estimation on KITTI

On KITTI Odometry [9] (Table 1), we additionally compare against two long-sequence-specialized baselines: the retrained-from-scratch LongStream [5] and the RL-policy-trained Keyframe-VO [7]. Concurrent inference-time methods TTSA3R [30] and MeMix [8] do not report KITTI results and are instead compared on the indoor benchmarks of Sec. 5.1. AFG-Pose achieves the lowest average ATE (43.96 m), reducing TTT3R by 35.9% and surpassing both long-sequence-specialized baselines—LongStream (51.90) and Keyframe-VO (87.00)—despite requiring no retraining, no architectural change, and no extra forward pass; AFG-Img also improves TTT3R to 56.60. Consistent with the EMA horizon analysis in Sec. 3, gains tend to grow with length rather than with loop structure: on the three sequences longer than 4000 frames (00/02/08) AFG-Pose reduces ATE over TTT3R by 39–51% and on 09 by 64%, while the shortest (04, $L = 271$) gets worse. The trend is not strict (05: +11%; 07: +45%), and loops do not separate the outcomes: the largest gain (09) and the largest loss (07) are both loop sequences, and among the three 1101-frame sequences the loop-free 01 (−17%) sits between the two loop sequences 06 (−20%) and 07 (+45%). The per-sequence breakdown is in Appendix A9. AFG-Pose beats AFG-Img on 7 of 11 sequences, matching the regime mapping of Sec. 4; trajectories for all 11 sequences are shown in Appendix A10.

Table 1: **Long-sequence pose estimation on KITTI [9].** ATE ↓ (m) of the aligned trajectory. Baseline results are taken from the original papers. Best in **bold**, second-best underlined.

Methods	KITTI (ATE ↓)											
	00 4541, 3.7km	01 1101, 2.5km	02 4661, 5.1km	03 801, 0.6km	04 271, 0.4km	05 2761, 2.2km	06 1101, 1.2km	07 1101, 0.7km	08 4071, 3.2km	09 1591, 1.7km	10 1201, 0.9km	Avg.
CUT3R	190.38	90.59	264.39	20.40	7.31	92.25	67.54	22.48	145.08	67.42	40.00	91.62
TTT3R	118.20	102.98	242.05	14.38	4.51	**31.62**	37.77	**12.00**	81.52	79.10	29.82	68.54
InfiniteVGGT	186.46	623.62	289.16	166.74	68.00	143.84	117.57	85.33	221.36	215.41	156.92	206.78
FastVGGT	102.50	176.80	170.00	11.20	2.50	76.10	131.20	61.50	99.30	99.30	36.70	87.90
StreamVGGT	191.93	653.06	303.35	157.50	108.24	160.46	133.71	89.00	263.95	216.69	209.80	226.15
LongStream	92.55	**46.01**	134.70	**3.81**	**1.95**	84.69	**23.12**	14.93	62.07	85.61	**21.48**	51.90
Keyframe-VO	138.10	179.10	153.50	12.00	13.20	131.60	73.00	45.90	86.50	97.50	26.80	87.00
AFG-Img **(Ours)**	89.03	79.40	229.25	14.68	8.47	32.21	31.29	14.01	49.61	47.21	27.44	56.60
AFG-Pose **(Ours)**	**72.22**	85.49	**118.98**	12.07	6.49	35.16	30.15	17.44	**45.92**	**28.56**	31.09	**43.96**

5.3 Video Depth Estimation

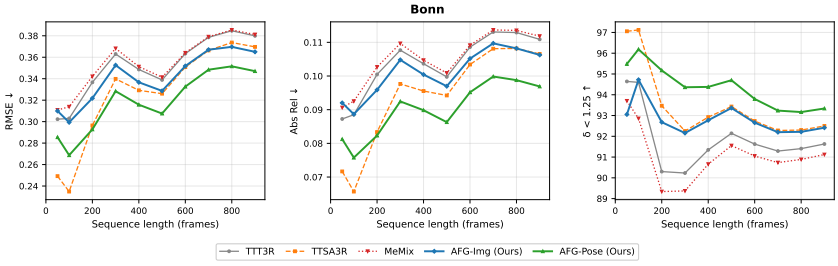


Figure 4: **Video depth estimation on Bonn (metric alignment).** For $L \geq 200$, AFG-Pose improves over TTT3R, TTSA3R, and MeMix on all metrics, with gains widening at longer sequences; at $L \leq 100$ TTSA3R is best.

Following prior work [22, 3, 8], we evaluate video depth estimation on Bonn [15], which covers dynamic indoor scenes. We report absolute relative error (Abs Rel), root mean square error (RMSE), and the percentage of predicted depths within a 1.25-factor of ground truth ($\delta < 1.25$) under metric alignment; results are shown in Fig. 4, with per-length numerical values under both metric and scale-and-shift alignments tabulated in Appendix A11. For $L \geq 200$, both AFG-Img and AFG-Pose improve depth estimation over TTT3R and MeMix, with the gains becoming more pronounced on longer sequences where recurrent state degradation accumulates; averaged over all lengths, AFG-Pose reduces AbsRel by 13.0% over TTT3R. AFG-Pose is the stronger variant throughout, while AFG-Img provides a smaller improvement. MeMix tracks TTT3R closely at all lengths. On short sequences ($L \leq 100$), where drift has not yet accumulated, TTSA3R is best; AFG-Pose still improves over TTT3R there, while AFG-Img is on par with or slightly behind it.

5.4 3D Reconstruction

Following common practice in long-horizon streaming reconstruction [22, 3, 8], we evaluate multiview reconstruction on 7-Scenes [17] and NRGBD [1] at three sequence lengths (300/400/500 frames) with sparse sampling (one frame every two frames, -S); the full per-length, per-baseline table is provided in Table 18 of Appendix A12. Full-attention models such as VGGT [21] run out of memory at all tested lengths. Among constant-memory recurrent baselines, reconstruction quality degrades consistently as the input horizon grows: TTT3R mean accuracy on 7-Scenes deteriorates from 0.039 at L=300 to 0.065 at L=500, and on NRGBD from 0.104 to 0.166, reflecting cumulative state drift under long sequences.

Our AFG variants substantially reduce this length-dependent degradation. On 7-Scenes, AFG-Img maintains near-constant accuracy across all lengths (0.023/0.023/0.024 at L=300/400/500), while TTT3R degrades by 67% over the same range. On NRGBD, where drift is more pronounced, AFG-Img reduces the mean Acc error by 24% relative to TTSA3R at L=500 (0.092 vs. 0.121) and cuts completeness error by 45% (0.027 vs. 0.049). The MeMix row applies MeMix to TTT3R, the setting proposed in its paper and the like-for-like comparison, since our variants also build on TTT3R. Against the strongest combination reported by MeMix, MeMix on TTSA3R (mean accuracy 0.033 on 7-Scenes and 0.114 on NRGBD at L=500, values from [8]; our TTT3R baseline matches theirs), AFG-Img still reaches 0.024 and 0.092. Qualitative comparisons (Fig. 7, Appendix A13) corroborate the numbers: without adaptive gating, recurrent baselines accumulate pose drift and produce fragmented geometry, while AFG yields more coherent surfaces and better-preserved scene structure across both datasets.

5.5 Ablation Studies and Analysis

Is adaptivity necessary, and which gate carries the gain? Table 2 answers both on TUM-RGBD pose and Bonn depth. *Fixed-α probe.* Replacing the adaptive gate with a constant $\alpha \in \{0.1, \ldots, 0.7\}$ already improves ATE over TTT3R, but ATE and rotation are best at opposite ends of the range: ATE is U-shaped with its minimum near $\alpha = 0.3$, while rotation error improves monotonically up to $\alpha = 0.7$ (1.849 $\rightarrow$ 0.429). Suppressing drift requires weak writes on redundant frames, whereas fine rotation requires strong writes, and a constant cannot tell the two kinds of frame apart; the adaptive gate improves on the best constant for ATE (α=0.3) by 18% and on the best constant for AbsRel (α=0.1, which costs 4$\times$ the rotation error) by 3%, while keeping rotation error close to TTT3R. *Write-schedule probe.* To test whether the gain comes merely from writing less, we let the state absorb only every k-th frame (every frame is still encoded, decoded and evaluated). Writing less helps pose (0.109 $\rightarrow$ 0.066 at k=3) but damages rotation and barely moves depth (0.098 vs. our 0.086 AbsRel); on KITTI, writing every third frame gives 53.12 against our 43.96. The rest of the gain comes from letting the write strength follow the content (see Appendix A5 for subsampled input). *Component probe.* Applying α alone to CUT3R (without β) already reaches ATE 0.063 and AbsRel 0.085, against 0.109 and 0.100 for TTT3R, showing the frame-level gate independently captures most of the gain; adding β further reduces ATE to 0.054 (14% additional) while maintaining depth quality, confirming β's complementary token-level selectivity—α controls *when* to update at the frame level, β controls *where* at the token level.

Robustness to the gate constants. Sweeping $\tau \in [0.5, 1.5]$ for AFG-Pose on TUM-RGBD pose and Bonn depth (Appendix A2.1), all configurations outperform TTT3R and ATE varies only within 0.051–0.064; sweeping the sharpness $\kappa \in \{1, 2, 3, 5\}$ gives ATE 0.066–0.052 (Appendix A2.2), so the method does not require per-scene tuning. Larger τ trades a small RPE_r increase for further ATE and AbsRel gains.

The rotation trade-off. At the default setting AFG-Pose raises RPE_r on TUM-RGBD (0.520 vs. 0.443) while halving ATE. The cost comes from the setting, not the method: at $\tau = 0.5$ it improves both metrics (0.064 ATE, 0.376 RPE_r; Appendix A2.1), and α alone gives the best rotation (0.376); we attribute the loss to frames where α and β are both low, which receive too little update for fine rotation. The default leans towards ATE.

Does the pose gate track the scene or the state? Since p_t is decoded from S_{t-1}, Δp_t could in principle reflect model instability rather than geometric novelty. On TUM `walking_xyz` we compare

Table 2: **Ablation: necessity of adaptivity and contribution of each gate.** TUM-RGBD (1000 frames, pose) and Bonn (500 frames, depth). Blocks sweep a constant α, a content-independent write schedule (the state absorbs every k-th frame), and the contribution of β on top of α. "–": not measured. Best in **bold**.

Configuration	α_t	β_t	Pose (TUM) ATE↓	Pose (TUM) RPE_r↓	Depth (Bonn) AbsRel↓	Depth (Bonn) δ<1.25↑
TTT3R	$\equiv 1$	✓	0.109	0.443	0.100	92.1
Fixed-α probe (does adaptivity matter?)						
$\alpha_t \equiv 0.1$	0.1	✓	0.082	1.849	0.089	–
$\alpha_t \equiv 0.2$	0.2	✓	0.075	1.078	0.094	–
$\alpha_t \equiv 0.3$	0.3	✓	0.066	0.854	0.095	93.2
$\alpha_t \equiv 0.5$	0.5	✓	0.079	0.449	0.099	92.8
$\alpha_t \equiv 0.7$	0.7	✓	0.093	0.429	0.100	92.3
Write-schedule probe (is it only writing less?)						
write every 2nd frame	$\{0, 1\}$	✓	0.079	0.439	0.100	91.9
write every 3rd frame	$\{0, 1\}$	✓	0.066	0.648	0.098	92.7
Component probe (does β help on top of α?)						
α only (no β)	adaptive		0.063	**0.376**	**0.085**	**95.1**
$\alpha + \beta$ (Ours, full)	adaptive	✓	**0.054**	0.520	0.086	94.7

α_t with two ground-truth quantities that do not depend on our model: the camera rotation since the previous frame, and the pose error at that frame after Sim(3) alignment. The gate follows the camera (Spearman $+0.50$) and not the error (-0.12), although the state is never accurate there (2.9 cm accumulated error on average). The coupling is not zero, but it does not drive the gate; moreover, $\alpha_t \leq 1$ means no frame is ever written more strongly than TTT3R would write it, and AFG-Img reads encoder features only and has no such coupling (Appendix A6).

Why not combine the two gates? Fusing α^{img} and α^{pose} (OR, AND, or a weighted average) never improves over the best single gate, since both draw on correlated per-frame novelty (Appendix A7); we therefore keep them as complementary regime-specific variants.

6 Conclusion

In this paper, we rethought why constant-memory recurrent 3D reconstructors drift on long sequences, and traced the failure to a structural blind spot in the per-token gate: averaged across thousands of cross-attention interactions, every gate collapses into nearly the same magnitude, leaving no way to distinguish a novel frame from a redundant one. To restore the missing axis, i.e., frame-level adaptivity, we introduced Adaptive Frame Gating, a parameter-free scalar gate computed at inference time from features the model already produces, with no retraining, no architectural change, and no extra forward pass. Across pose, depth, and reconstruction tasks, AFG outperforms inference-time gating baselines on long sequences, and on long outdoor trajectories surpasses methods purpose-built for long sequences.

Limitations and future work. AFG closes the frame-level axis but does not address every long-sequence failure mode of fixed-state recurrent models, most notably, state saturation under sustained novelty. Its benefit is tied to accumulated drift: on short sequences it gives little and can be worse (KITTI `04`; also `05`, `07`), since the gate can only protect information that would otherwise have been overwritten. At the default setting it trades some rotation accuracy for lower drift (Sec. 5.5). Because $\alpha_t \leq 1$, AFG only weakens writes; amplifying highly novel frames destabilises the frozen model outdoors (Appendix A3).

Acknowledgments and Disclosure of Funding

This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grants No. 62525106 and No. 62472046, and in part by the Young Elite Scientists Sponsorship Program of the Beijing High Innovation Plan under Grant No. 20250866. The authors declare no competing interests.

References

[1] Dejan Azinović, 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*, pages 6290–6301, 2022.

[2] Carlos Campos, Richard Elvira, Juan J Gómez Rodríguez, José MM 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.

[3] Xingyu Chen, Yue Chen, Yuliang Xiu, Andreas Geiger, and Anpei Chen. Ttt3r: 3d reconstruction as test-time training. *arXiv preprint arXiv:2509.26645*, 2025.

[4] 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*, pages 5273–5284, 2025.

[5] Chong Cheng, Xianda Chen, Tao Xie, Wei Yin, Weiqiang Ren, Qian Zhang, Xiaoyang Guo, and Hao Wang. Longstream: Long-sequence streaming autoregressive visual geometry. *arXiv preprint arXiv:2602.13172*, 2026.

[6] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5828–5839, 2017.

[7] Weichen Dai, Wenhan Su, Da Kong, Yuhang Ming, and Wanzeng Kong. Keyframe-based feed-forward visual odometry. *arXiv preprint arXiv:2601.16020*, 2026.

[8] Jiacheng Dong, Huan Li, Sicheng Zhou, Wenhao Hu, Weili Xu, and Yan Wang. Memix: Writing less, remembering more for streaming 3d reconstruction. *arXiv preprint arXiv:2603.15330*, 2026.

[9] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The international journal of robotics research*, 32(11):1231–1237, 2013.

[10] 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.

[11] Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3d with mast3r. In *European conference on computer vision*, pages 71–91. Springer, 2024.

[12] 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. *arXiv preprint arXiv:2509.05296*, 2025.

[13] 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.

[14] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. Orb-slam: A versatile and accurate monocular slam system. *IEEE transactions on robotics*, 31(5):1147–1163, 2015.

[15] Emanuele Palazzolo, Jens Behley, Philipp Lottes, Philippe Giguere, and Cyrill Stachniss. Refusion: 3d reconstruction in dynamic environments for rgb-d cameras exploiting residuals. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7855–7862. IEEE, 2019.

[16] You Shen, Zhipeng Zhang, Yansong Qu, Xiawu Zheng, Jiayi Ji, Shengchuan Zhang, and Liujuan Cao. Fastvggt: Training-free acceleration of visual geometry transformer. *arXiv preprint arXiv:2509.02560*, 2025.

[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*, pages 2930–2937, 2013.

[18] Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. A benchmark for the evaluation of rgb-d slam systems. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pages 573–580. IEEE, 2012.

[19] Yu Sun, Xinhao Li, Karan Dalal, Chloe Hsu, Sanmi Koyejo, Carlos Guestrin, Xiaolong Wang, Tatsunori Hashimoto, and Xinlei Chen. Learning to (learn at test time). *arXiv preprint arXiv:2310.13807*, 2023.

[20] Hengyi Wang and Lourdes Agapito. 3d reconstruction with spatial memory. In *2025 International Conference on 3D Vision (3DV)*, pages 78–89. IEEE, 2025.

[21] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 5294–5306, 2025.

[22] Qianqian Wang, Yifei Zhang, Aleksander Holynski, Alexei A Efros, and Angjoo Kanazawa. Continuous 3d perception model with persistent state. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 10510–10522, 2025.

[23] 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*, pages 20697–20709, 2024.

[24] Yuqi Wu, Wenzhao Zheng, Jie Zhou, and Jiwen Lu. Point3r: Streaming 3d reconstruction with explicit spatial pointer memory. *arXiv preprint arXiv:2507.02863*, 2025.

[25] Tao Xie, Peishan Yang, Yudong Jin, Yingfeng Cai, Wei Yin, Weiqiang Ren, Qian Zhang, Wei Hua, Sida Peng, Xiaoyang Guo, et al. Scal3r: Scalable test-time training for large-scale 3d reconstruction. *arXiv preprint arXiv:2604.08542*, 2026.

[26] Lanbo Xu, Liang Guo, Caigui Jiang, and Cheng Wang. Pas3r: Pose-adaptive streaming 3d reconstruction for long video sequences. *arXiv preprint arXiv:2603.21436*, 2026.

[27] 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 Computer Vision and Pattern Recognition Conference*, pages 21924–21935, 2025.

[28] 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.

[29] 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.

[30] Zhijie Zheng, Xinhao Xiang, and Jiawei Zhang. Ttsa3r: Training-free temporal-spatial adaptive persistent state for streaming 3d reconstruction. *arXiv preprint arXiv:2601.22615*, 2026.

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

Appendix

A1 Redundancy-Injection Probe: Mechanism Verification

We test the central causal claim of the main text—that the structural constancy of β produces a short memory horizon, and the resulting long-sequence drift is suppressed by α—under a controlled stress test that provides ground-truth zero-information frames and isolates drift from real motion: 100 pixel-identical frames injected into a real sequence with the GT camera held static.

Setup. We take the first 500 frames of TUM-RGBD `walking_xyz` as a normal information stream, then append 100 pixel-identical copies of frame 499 at positions 500–599, with the GT camera pose held constant on the injected segment. The same frozen CUT3R checkpoint is run with three update rules: TTT3R (Eq. 2), AFG-Img, and AFG-Pose (Eqs. 5, 6). For each variant we record the per-token gate β_t, the frame-level gate α_t (where applicable), the per-step state update $\|\Delta S_t\|$, and the estimated camera trajectory. Estimated trajectories are aligned to GT with Sim(3) Umeyama prior to error computation, following the convention of Sec. 5.

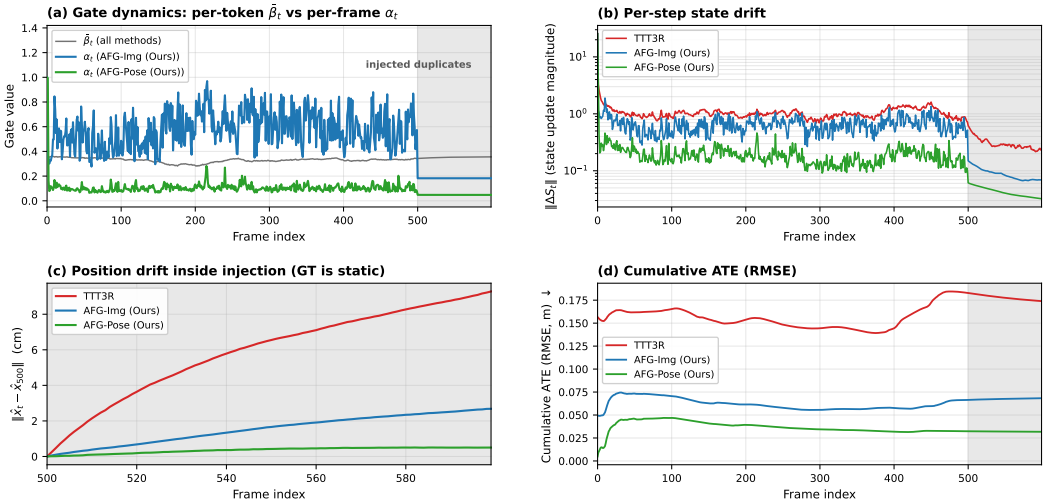


Figure 5: **Redundancy-injection probe on TUM `walking_xyz`.** Frames 0–499 are the original sequence; frames 500–599 (gray shading) are 100 pixel-identical copies of frame 499 with GT camera held static. **(a)** The per-token gate $\bar{\beta}_t$ (gray; identical across methods) remains flat at $\sim$0.35 throughout, including the injected segment, confirming its content-independence; the per-frame gate α_t drops sharply on redundant frames (AFG-Img to 0.18, AFG-Pose to 0.048). **(b)** The per-step state update $\|\Delta S_t\|$ (log scale): TTT3R retains a non-negligible $\sim$0.31 even on identical inputs, while AFG-Pose suppresses it to 0.043 (a 7$\times$ reduction). **(c)** Position drift inside the injection, $\|\hat{x}_t - \hat{x}_{500}\|$: since GT is static, any non-zero value is pure drift. TTT3R drifts $\sim$9 cm; AFG-Pose stays below 0.5 cm (18$\times$ reduction). **(d)** Sim(3)-aligned cumulative ATE (RMSE): TTT3R's curve visibly bends upward on the injected segment while AFG-Pose remains flat. $\bar{\beta}_t$ and α_t are not comparable in absolute value (different gate types); the diagnostic signal is how each *responds* to redundancy. Colors: TTT3R (red), AFG-Img (blue), AFG-Pose (green), shared $\bar{\beta}_t$ (gray).

β is content-blind, even in the worst case. Panel (a) of Fig. 5 shows $\bar{\beta}_t$ holding at 0.352 on the 100 injected duplicates, statistically indistinguishable from its value on the preceding informative frames. This is a stronger statement than the dataset-level statistics of Sec. 3.2: even when the input carries strictly zero new information, the per-token gate does not adapt. The architectural origin of bounded β identified in Sec. 3.2 is therefore not a statistical artifact of typical sequences but a structural property of the gating mechanism itself.

AFG suppresses redundant updates by an order of magnitude. Panel (b) shows that TTT3R's per-step update naturally decreases on identical inputs (because ΔS_t shrinks when the input residual

is small) but plateaus at $\|\Delta S_t\| \approx 0.31$—still large enough to materially perturb the state. AFG-Pose, by additionally gating with $\alpha_t \to 0.048$, drives the effective update down to 0.043, a further $7\times$ suppression on top of TTT3R's intrinsic decay.

Geometric drift is reduced by an order of magnitude. Because the GT camera is held static on $[500, 599]$, the quantity $\|\hat{x}_t - \hat{x}_{500}\|$ in panel (c) is pure drift introduced by the recurrent update. TTT3R drifts to $\sim$9 cm despite no real motion; AFG-Pose stays under 0.5 cm—an $18\times$ reduction. Panel (d) shows the same effect at trajectory scale: TTT3R's cumulative ATE (RMSE) bends upward on the injected segment, while AFG-Pose remains flat.

Numerical closure with the main-text horizon prediction. Substituting the measured $\alpha_{\min} = 0.048$ and $\bar{\beta} = 0.352$ from panel (a) into the AFG horizon formula of Sec. 4 yields

$$H_{\text{AFG}} \;=\; \frac{1}{\alpha_{\min}\,\bar{\beta}} \;\approx\; 60 \text{ frames},$$

while measuring the decay of the drift directly on the injected segment gives $\sim$64 frames. Two numbers reached by different routes agree, which makes the mechanism checkable rather than merely plausible. The TTT3R horizon under the same $\bar{\beta}$ is $1/\bar{\beta} \approx 2.8$ frames, consistent with the $\sim$3 frames of Sec. 3.3. The structural mechanism diagnosed in Sec. 3 and the remedy proposed in Sec. 4 are thus both quantitatively confirmed on a single controlled experiment.

A2 Gate Constants and Sensitivity

Both gates share the form $\alpha_t = \sigma(\kappa(\|\Delta f_t\|_2 - \tau))$ (Eqs. 5, 6); Table 3 lists the constants, fixed across all benchmarks. A sigmoid on a raw distance needs a centre and a width: τ is the centre, κ the sharpness, and the two features live on different scales. The value $\alpha_t \approx 0.09$ shown in Fig. 1 (AFG-Pose) corresponds to $\|\Delta p_t\|_2 \approx 0.23$.

Table 3: **Gate constants.** The lowest value is attained at zero feature change.

Variant	Feature	(κ, τ)	Lowest value $\sigma(-\kappa\tau)$
AFG-Img	encoder global feature g_t	$(1, 1.5)$	0.18
AFG-Pose	decoder pose token p_t	$(3, 1.0)$	0.047

A2.1 Threshold τ

We sweep the threshold τ of the AFG-Pose gate $\alpha_t = \sigma(\kappa(\|\Delta p_t\|_2 - \tau))$, with $\kappa = 3$ fixed, across $\{0.5, 0.75, 1.0, 1.25, 1.5\}$ on TUM-RGBD pose and Bonn depth (Table 4). Increasing τ progressively improves ATE and AbsRel at the cost of a higher RPE_r, reflecting a trade-off between global trajectory accuracy and local rotational smoothness. All swept configurations significantly outperform TTT3R (ATE = 0.109, AbsRel = 0.100), and ATE varies only within 0.051–0.064, confirming the method does not require per-scene tuning. We adopt $\tau = 1.0$ as the AFG-Pose default in all main-paper experiments. At $\tau = 0.5$ AFG-Pose improves over TTT3R on both ATE and RPE_r (0.064/0.376 vs. 0.109/0.443), showing that the rotation cost at the default is a choice of operating point rather than a property of the gate.

Table 4: **Threshold sensitivity (AFG-Pose, $\kappa = 3$).** Performance varying τ on TUM-RGBD (1000 frames, pose) and Bonn (500 frames, depth). All configurations outperform TTT3R.

	Pose (TUM)		Depth (Bonn)	
τ	ATE↓	RPE_r↓	AbsRel↓	δ<1.25↑
0.5	0.064	0.376	0.093	94.2
0.75	0.058	0.391	0.090	94.7
1.0 (default)	0.054	0.520	0.086	94.7
1.25	0.051	0.603	0.084	94.7
1.5	0.058	0.751	0.082	94.7

A2.2 Sharpness κ

Table 5 sweeps κ for AFG-Pose with $\tau = 1.0$. Every setting improves substantially over TTT3R; even $\kappa = 1$, i.e. the gate without a sharpness constant, reduces ATE from 0.109 to 0.066 and RPE_r from 0.443 to 0.421. Larger κ trades rotation for drift, mirroring the τ sweep above.

Table 5: **Sharpness sweep (AFG-Pose, $\tau = 1.0$).** TUM-RGBD (1000 frames) and Bonn (500 frames). "–": not measured.

κ	TUM ATE↓	TUM RPE_r↓	Bonn AbsRel↓
TTT3R (no frame gate)	0.109	0.443	0.100
1	0.066	0.421	0.095
2	0.059	–	–
3 (default)	0.054	0.520	0.086
5	**0.052**	–	–

A3 Capping the Frame Gate at 1

Since α_t acts as an inference-time learning rate, values above 1 could in principle help highly novel frames. We amplify only the frames the gate already treats as novel: $\alpha_t = 1$ is remapped to a cap s, while frames at or below the threshold are left untouched. Indoor results use the protocol of Table 2; KITTI uses all 11 sequences under the protocol of Table 1.

Table 6: **Raising the cap of the frame gate (AFG-Pose).**

Cap s	TUM ATE↓	TUM RPE_r↓	Bonn AbsRel↓	KITTI ATE↓ (mean of 11)
TTT3R (no frame gate)	0.109	0.443	0.100	68.54
$s = 1$ (Ours)	**0.054**	0.520	0.086	**43.96**
$s = 2$	0.055	0.478	0.083	55.88
$s = 5$	0.058	**0.411**	**0.078**	97.71

Indoors, raising the cap changes little: ATE rises slightly while rotation and depth improve. On KITTI the error rises steadily with the cap; at $s = 5$ it is worse than using no frame gate at all on 9 of the 11 sequences, and on sequence `09` the capped gate is 64% better than TTT3R while $s = 5$ is 583% worse. The difference lies in how the gate values are distributed: indoors only a few percent of frames clear the threshold, so amplifying them touches almost nothing, whereas in fast outdoor motion a much larger share clears it, and updates larger than anything seen during training accumulate until the state degrades. Scaling the whole gate, which raises its minimum as well as its cap, is worse still (TUM ATE $0.054 \rightarrow 0.060$ for AFG-Pose, $0.084 \rightarrow 0.112$ for AFG-Img). The cap therefore keeps AFG safe on a frozen checkpoint: its worst case is exactly TTT3R's own update.

A4 Comparison with Keyframe-Style Gating

To separate AFG from keyframe selection at the level of mechanism (Sec. 4.3), we vary one choice at a time while keeping the feature and threshold of our gate. *Hard vs. graded*: we binarise the gate, $\alpha_t = 1$ above the threshold and a fixed low value otherwise. *Dropping vs. keeping*: the low value is set to 0.05 (every frame keeps writing weakly) or to 0 (the classical admit/skip rule). *Reference frame*: the feature change is measured against the last accepted frame (reset whenever the gate opens) instead of the previous frame, closer to classical keyframe criteria. All runs use the frozen checkpoint and the protocol of Table 2.

Dropping non-keyframes is the most harmful choice: a zero-weight frame can only be recovered by later frames writing more, whereas with a positive minimum an under-rated frame is merely delayed. On TUM the hard and graded responses tie on ATE but the graded one is better on rotation, and on ScanNet the graded gate is clearly better for both variants. Measuring novelty against the last accepted frame is worse on both pose and depth, so the adjacent-frame change is not only the simpler signal but the better one for this update rule.

Table 7: **Keyframe-style variants of the frame gate.** TUM-RGBD (1000 frames), ScanNet (1000 frames), Bonn (500 frames). "–": not measured.

Variant	TUM (AFG-Pose) ATE↓	TUM (AFG-Pose) RPE_r↓	ScanNet ATE↓ AFG-Img	ScanNet ATE↓ AFG-Pose	Bonn AbsRel↓
TTT3R (no frame gate)	0.109	0.443	0.405		0.100
binarised, low value 0 (admit/skip)	0.061	0.880	–	–	–
binarised, low value 0.05	**0.052**	0.647	0.300	0.352	–
graded, last accepted frame as reference	0.065	–	–	–	0.092
graded, previous frame (Ours)	0.054	**0.520**	**0.282**	**0.281**	**0.086**

A5 Is the Gain Only from Writing Less?

Write schedule on full-rate input. Table 2 compares AFG against content-independent write schedules, in which every frame is still encoded, decoded and evaluated but the state absorbs only every k-th frame. Changing only how often the state is written gives, on KITTI (mean ATE over 11 sequences), 68.54 when writing every frame (TTT3R), 53.12 when writing every third frame, and 43.96 when the content sets the write strength (AFG-Pose). Writing less helps pose both indoors and outdoors, but the gate lowers the error by a further 18% on TUM ($0.066 \rightarrow 0.054$) and 17% on KITTI, and on depth nearly all of the gain comes from content (Table 2).

Subsampled input. We additionally subsample the *input* stream, feeding only every third frame ($k = 3$) to both TTT3R and AFG-Pose, so that both models see exactly the same frames and are scored at exactly the same points; the ScanNet benchmark is already built on a stride-3 stream. Table 8 shows that on subsampled input the frame gate still improves depth, KITTI and ScanNet, but is worse on TUM (ATE $+8\%$, RPE_r $+22\%$). Subsampling and gating act on different quantities (how many frames enter the model vs. how strongly each writes into the state), and their gains combine on most benchmarks but not on all. On the same ScanNet stream, the per-token methods TTSA3R (0.388) and MeMix (0.367) stay close to TTT3R, so a fixed stride combined with token-level adaptivity does not replace frame-level control.

Table 8: **Input subsampling (stride $k = 3$), same frames for both methods.** ScanNet is natively evaluated on a stride-3 stream.

Benchmark / metric	TTT3R	AFG-Pose
TUM, ATE↓	**0.0604**	0.0650
TUM, RPE_r↓	**0.934**	1.137
Bonn, AbsRel↓	0.0978	**0.0861**
Bonn, δ<1.25↑	92.5	**94.3**
KITTI, ATE↓ (mean of 11)	106.79	**95.69**
ScanNet, ATE↓	0.405	**0.281**

The gain on subsampled input also grows with length. On the four longest KITTI sequences at stride 3 (Table 9), AFG-Pose lowers the mean ATE by 21.6%.

Table 9: **Four longest KITTI sequences at input stride $k = 3$.** ATE↓ (m).

Sequence	Frames	TTT3R	AFG-Pose
02	4661	241.25	**208.59**
00	4541	178.61	**117.74**
08	4071	168.44	**117.21**
05	2761	121.37	**112.97**
Mean		177.42	**139.13** (-21.6%)

A6 Coupling Between the Pose Gate and the Recurrent State

The pose token p_t is decoded from the current image and the state S_{t-1}, so Δp_t measures how much a frame changes the model's pose inference, and could in principle reflect a drifting state rather than geometric novelty. If the gate were driven by the model's own error, it should open widest where that error is largest. On TUM `walking_xyz` we record α_t for every frame together with two ground-truth quantities that do not rely on our model: the camera rotation since the previous frame, and the pose error at that frame (position gap after Sim(3) alignment).

- **Correlation.** Over all frames, α_t has a Spearman rank correlation of $+0.50$ with the camera rotation and -0.12 with the pose error. Rank correlation is used because the gate is heavy-tailed.
- **Extremes.** The top 10% of frames by α_t carry $1.68\times$ the camera rotation of the bottom 50% (0.82° vs. 0.49° per frame), while their pose error is level, if anything lower (2.92 vs. 3.07 cm).
- **The state is not accurate.** The accumulated pose error on this sequence is 2.9 cm on average and 6.6 cm at worst, so the absence of a link is not due to an error-free state.

Three qualifications apply. (i) We do not claim the coupling is zero: that p_t reads the state is an architectural fact; our claim is that it does not drive the gate. (ii) Whether the state accumulates error is a property of the frozen backbone; AFG only rescales each frame's write, and with $\alpha_t \leq 1$ no frame is ever written more strongly than TTT3R would write it, whatever the state. (iii) AFG-Img reads encoder features only and never touches the state, so this coupling has no path there; the two variants nevertheless agree in trend on all six benchmarks.

A7 Combining the Two Gates

A natural question is whether AFG-Img and AFG-Pose can be combined. We evaluate three fusion strategies in Table 10: OR via $\max(\alpha^{\text{img}}, \alpha^{\text{pose}})$; multiplicative AND $\alpha^{\text{img}} \cdot \alpha^{\text{pose}}$ with relaxed thresholds; and a signal-strength weighted average. No fusion strategy improves over the best single gate. The OR variant lets too many redundant frames through (TUM ATE 0.079 vs. AFG-Pose 0.054), undoing selectivity. The multiplicative variant cuts ATE moderately (0.075) but explodes the rotation error (RPE$_r$ 1.137, $2.6\times$ TTT3R) by zeroing informative residuals when both gates fire low. The weighted variant matches AFG-Pose within noise (ATE 0.054, AbsRel 0.086): both gates draw from the same per-frame backbone activations and carry correlated novelty, so fusion adds no new selectivity. We therefore adopt the two as *complementary regime-specific variants* rather than fusing them.

Table 10: **Combining the two gates is not beneficial.** Fusion strategies for α^{img} and α^{pose} on TUM-RGBD (1000 frames, pose) and Bonn (500 frames, depth). No combination strictly improves over the best single gate. Best in **bold**.

Configuration	Pose (TUM)		Depth (Bonn)		
	ATE↓	RPE$_r$↓	AbsRel↓	RMSE↓	δ<1.25↑
TTT3R	0.109	0.443	0.100	0.339	92.1
AFG-Img alone	0.084	**0.411**	0.097	0.329	93.4
AFG-Pose alone	**0.054**	0.520	**0.086**	**0.308**	**94.7**
$\max(\alpha^{\text{img}}, \alpha^{\text{pose}})$ (OR)	0.079	0.414	0.095	0.323	94.0
$\alpha^{\text{img}} \cdot \alpha^{\text{pose}}$ (retuned)	0.075	1.137	0.089	0.315	94.1
weighted$(\alpha^{\text{img}}, \alpha^{\text{pose}})$	**0.054**	0.522	**0.086**	**0.308**	**94.7**

A8 Detailed Per-Length Pose Numbers

We report the per-length numerical values underlying Fig. 3 of Sec. 5.1. Tables 11 and 12 list ATE on TUM-RGBD and ScanNet across sequence lengths from 50 to 1000 frames, for TTT3R [3], TTSA3R [30], MeMix [8], and our two variants.

Table 11: **Detailed pose ATE on TUM-RGBD across sequence lengths.** ATE ↓ (m) under Sim(3) Umeyama alignment; values averaged over sequences. Best in **bold**.

Method	Sequence Length L									
	50	100	200	300	400	500	600	700	800	1000
TTT3R [3]	0.014	0.020	0.041	0.045	0.051	0.067	0.074	0.085	0.091	0.109
TTSA3R [30]	0.013	**0.017**	0.030	0.038	0.043	0.056	0.062	0.070	0.076	0.093
MeMix [8]	0.014	0.020	0.033	0.042	0.047	0.060	0.068	0.074	0.085	0.101
AFG-Img **(Ours)**	**0.012**	**0.017**	0.027	0.033	0.037	0.044	0.051	0.058	0.066	0.084
AFG-Pose **(Ours)**	0.013	**0.017**	**0.023**	**0.028**	**0.032**	**0.033**	**0.036**	**0.040**	**0.045**	**0.054**

Table 12: **Detailed pose ATE on ScanNet across sequence lengths.** ATE ↓ (m) under Sim(3) Umeyama alignment. Best in **bold**.

Method	Sequence Length L									
	50	100	200	300	400	500	600	700	800	1000
TTT3R [3]	0.033	0.072	0.144	0.192	0.236	0.278	0.315	0.345	0.371	0.405
TTSA3R [30]	**0.032**	**0.063**	**0.122**	0.169	0.213	0.259	0.298	0.329	0.356	0.388
MeMix [8]	0.036	0.075	0.147	0.194	0.231	0.261	0.288	0.308	0.330	0.367
AFG-Img **(Ours)**	0.033	0.071	0.135	**0.165**	**0.194**	**0.212**	**0.231**	**0.249**	**0.265**	0.282
AFG-Pose **(Ours)**	0.038	0.084	0.173	0.206	0.240	0.250	0.257	0.267	0.277	**0.281**

A9 Per-Sequence Gains on KITTI

Table 13 re-orders the KITTI results of Table 1 by sequence length and reports the relative change of AFG-Pose over TTT3R. The largest improvements occur on the longest sequences (all three sequences above 4000 frames improve by 39–51%), and the shortest sequence (`04`) gets worse. The relation is a trend rather than a rule: `05` and `07` degrade despite their length. AFG has no loop-closure component, and loop structure does not separate the outcomes (Sec. 5.2). The gate can only protect information that would otherwise have been overwritten, and on a short trajectory little has been; we list this as a limitation.

Table 13: **KITTI ATE (m) ordered by sequence length.** Change is relative to TTT3R; negative is better.

Seq.	Frames	TTT3R	AFG-Pose	Change	AFG-Img	Change
02	4661	242.05	118.98	-50.8%	229.25	-5.3%
00	4541	118.20	72.22	-38.9%	89.03	-24.7%
08	4071	81.52	45.92	-43.7%	49.61	-39.1%
05	2761	31.62	35.16	$+11.2\%$	32.21	$+1.9\%$
09	1591	79.10	28.56	-63.9%	47.21	-40.3%
10	1201	29.82	31.09	$+4.3\%$	27.44	-8.0%
01	1101	102.98	85.49	-17.0%	79.40	-22.9%
06	1101	37.77	30.15	-20.2%	31.29	-17.2%
07	1101	12.00	17.44	$+45.3\%$	14.01	$+16.8\%$
03	801	14.38	12.07	-16.1%	14.68	$+2.1\%$
04	271	4.51	6.49	$+43.9\%$	8.47	$+87.8\%$

A10 KITTI Trajectory Visualizations

Fig. 6 visualizes the recovered camera trajectories on all 11 KITTI Odometry sequences (`00`–`10`) for ground truth, TTT3R [3], and our AFG-Pose variant (labeled "Ours"), corresponding to the per-sequence ATE numbers in Table 1. Each estimated trajectory is aligned to ground truth by Sim(3) Umeyama; panel titles report the official KITTI path length. On most long sequences (`00`, `01`, `02`, `06`, `08`, `09`; ≥ 1200 m), TTT3R drifts visibly while Ours tracks the ground-truth route more closely, directly visualizing the long-sequence drift suppression; on `05` the two are comparable (ATE 35.16 vs. 31.62). On short sequences (`03`, `04`, `07`, `10`; ≤ 920 m), both methods track ground truth closely, consistent with the observation that the gating advantage emerges as accumulated drift grows.

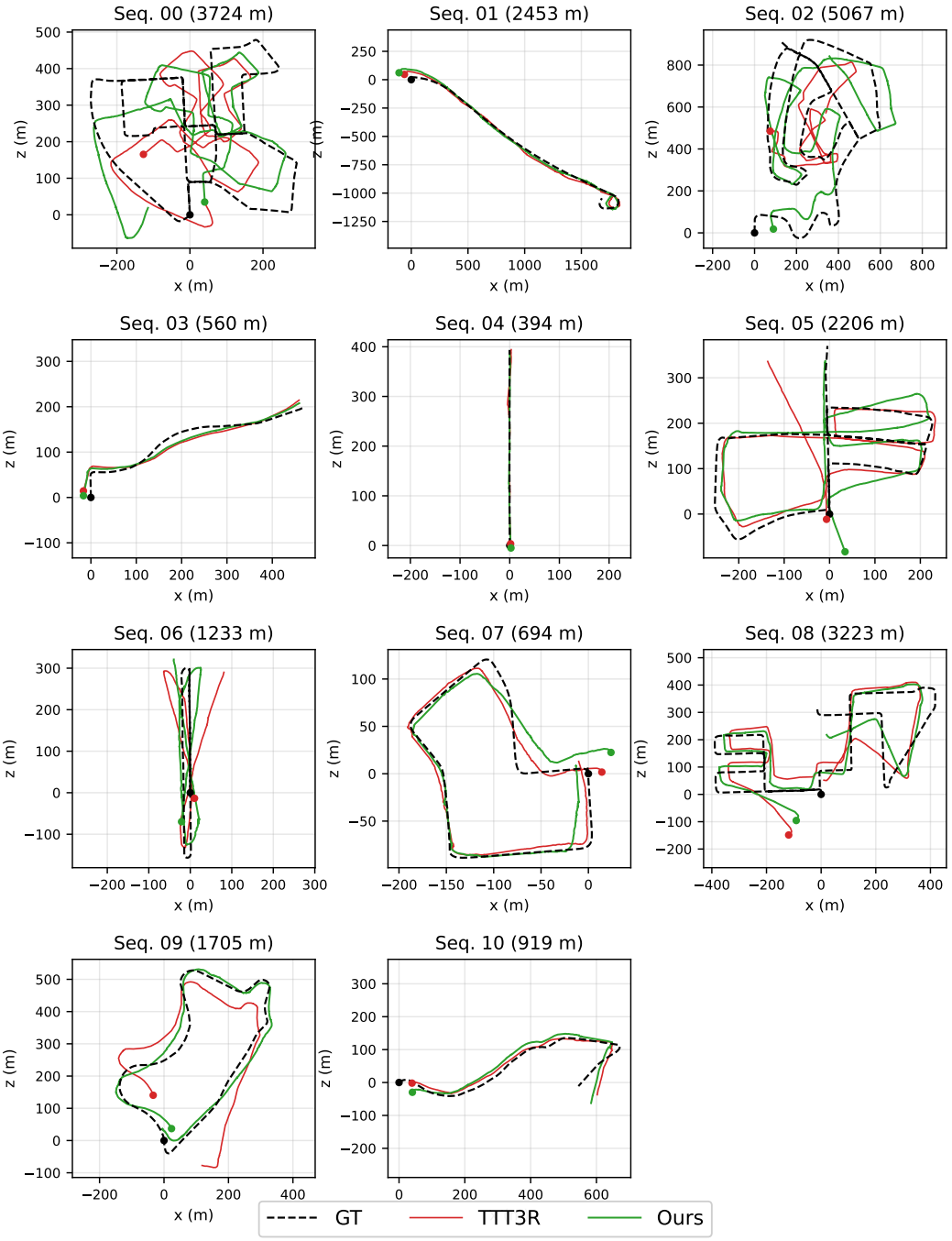


Figure 6: **Camera trajectory visualizations on KITTI Odometry.** Top-down (x–z) views of the recovered trajectories on all 11 sequences (00–10). Each estimated trajectory is aligned to ground truth via Sim(3) Umeyama; panel titles list the official KITTI path length in meters. On most long sequences ($\geq$ 1200 m), TTT3R [3] drifts substantially while Ours (AFG-Pose) tracks the ground-truth route more closely. On short sequences ($\leq$ 920 m), both methods track ground truth accurately, consistent with our finding that the gating advantage strengthens as accumulated drift grows.

A11 Detailed Bonn Depth Numbers

We provide the per-length numerical values plotted in Fig. 4 of Sec. 5.3 under metric alignment (Sec. A11.1), and additionally report results under scale-and-shift alignment (Sec. A11.2) as a robustness check, since scale-and-shift is the standard alignment in the depth-estimation literature. Across both alignments and all three metrics, AFG-Pose is the strongest variant for $L \geq 200$; TTSA3R wins at very short sequences ($L \leq 100$) where short-horizon visual matching dominates and drift has not yet accumulated.

A11.1 Metric Alignment

Tables 14, 15, and 16 list AbsRel, RMSE, and δ<1.25 under metric alignment, the strictest evaluation (no scale or shift compensation), corresponding directly to the curves of Fig. 4.

Table 14: **Bonn depth (Abs Rel ↓) under metric alignment, across sequence lengths.** Best in **bold**.

Method	Sequence Length L									
	50	100	200	300	400	500	600	700	800	900
TTT3R [3]	0.0872	0.0885	0.1005	0.1077	0.1036	0.0997	0.1086	0.1131	0.1129	0.1109
TTSA3R [30]	**0.0717**	**0.0657**	0.0834	0.0977	0.0955	0.0942	0.1034	0.1081	0.1082	0.1066
MeMix [8]	0.0906	0.0925	0.1027	0.1097	0.1047	0.1009	0.1091	0.1137	0.1135	0.1118
AFG-Img **(Ours)**	0.0920	0.0887	0.0959	0.1048	0.1004	0.0970	0.1052	0.1097	0.1082	0.1063
AFG-Pose **(Ours)**	0.0812	0.0758	**0.0823**	**0.0925**	**0.0898**	**0.0863**	**0.0952**	**0.0998**	**0.0988**	**0.0969**

Table 15: **Bonn depth (RMSE ↓) under metric alignment, across sequence lengths.** Best in **bold**.

Method	Sequence Length L									
	50	100	200	300	400	500	600	700	800	900
TTT3R [3]	0.302	0.303	0.337	0.363	0.349	0.339	0.363	0.379	0.385	0.380
TTSA3R [30]	**0.249**	**0.235**	0.296	0.340	0.329	0.326	0.351	0.366	0.374	0.370
MeMix [8]	0.311	0.314	0.342	0.368	0.351	0.341	0.364	0.379	0.385	0.381
AFG-Img **(Ours)**	0.310	0.299	0.322	0.353	0.337	0.329	0.352	0.367	0.370	0.365
AFG-Pose **(Ours)**	0.285	0.269	**0.293**	**0.328**	**0.316**	**0.308**	**0.333**	**0.348**	**0.352**	**0.347**

Table 16: **Bonn depth (δ<1.25 ↑, %) under metric alignment, across sequence lengths.** Best in **bold**.

Method	Sequence Length L									
	50	100	200	300	400	500	600	700	800	900
TTT3R [3]	94.6	94.6	90.3	90.2	91.3	92.1	91.6	91.3	91.4	91.6
TTSA3R [30]	**97.1**	**97.1**	93.5	92.3	92.9	93.4	92.7	92.3	92.3	92.5
MeMix [8]	93.7	92.9	89.3	89.4	90.7	91.6	91.0	90.7	90.9	91.1
AFG-Img **(Ours)**	93.1	94.7	92.7	92.2	92.8	93.4	92.7	92.2	92.2	92.4
AFG-Pose **(Ours)**	95.5	96.2	**95.2**	**94.4**	**94.4**	**94.7**	**93.8**	**93.2**	**93.2**	**93.3**

A11.2 Scale-and-Shift Alignment

Table 17 reports AbsRel under scale-and-shift alignment, the standard convention where each predicted depth map is fitted to ground truth via a per-frame scale and shift before evaluation. Patterns are consistent with the metric-alignment results above, confirming that the gains are not artifacts of the alignment protocol.

A12 Detailed 3D Reconstruction Numbers

We provide the per-length, per-baseline reconstruction metrics on 7-Scenes [17] and NRGBD [1] that underlie the analysis in Sec. 5.4. Table 18 reports Accuracy, Completeness, and Normal Consistency (mean and median) at three sequence lengths $L \in \{300, 400, 500\}$, comparing TTT3R [3], MeMix [8],

Table 17: **Bonn depth (Abs Rel ↓) under scale-and-shift alignment, across sequence lengths.** Best in **bold**.

Method	Sequence Length L									
	50	100	200	300	400	500	600	700	800	900
TTT3R [3]	0.0699	0.0608	0.0636	0.0751	0.0747	0.0720	0.0833	0.0899	0.0909	0.0894
TTSA3R [30]	**0.0575**	**0.0524**	0.0684	0.0779	0.0765	0.0749	0.0854	0.0917	0.0926	0.0912
MeMix [8]	0.0716	0.0624	0.0630	0.0749	0.0746	0.0721	0.0827	0.0894	0.0902	0.0888
AFG-Img **(Ours)**	0.0646	0.0557	0.0593	0.0691	0.0683	0.0665	0.0770	0.0841	0.0842	0.0829
AFG-Pose **(Ours)**	0.0618	0.0542	**0.0585**	**0.0675**	**0.0674**	**0.0654**	**0.0758**	**0.0821**	**0.0821**	**0.0808**

TTSA3R [30], and our two variants; full-attention VGGT [21] runs out of memory at all tested lengths. Both AFG variants substantially improve over the inference-time baselines, with the gap widening at longer sequences.

Table 18: **3D reconstruction on 7-Scenes [17] and NRGBD [1].** Models are evaluated with sparse sampling (one frame every two frames, -S). All recurrent rows use the same frozen CUT3R checkpoint; MeMix is applied to TTT3R (the setting of its paper), and TTSA3R is run by us. Shaded rows are our methods. **Bold** indicates the best result in each column at the same input length.

		7-Scenes-S						**NRGBD-S**					
		Acc ↓		Comp ↓		NC ↑		Acc ↓		Comp ↓		NC ↑	
Model	L	Mean	Med.	Mean	Med.	Mean	Med.	Mean	Med.	Mean	Med.	Mean	Med.
VGGT [21]	300/400/500	*OOM at all tested lengths*											
TTT3R [3]	300	0.039	0.023	0.023	0.005	0.565	0.600	0.104	0.045	0.026	0.005	0.605	0.668
	400	0.050	0.029	0.026	0.005	0.557	0.587	0.144	0.066	0.068	0.009	0.592	0.643
	500	0.065	0.037	0.031	0.006	0.550	0.575	0.166	0.087	0.087	0.016	0.588	0.637
MeMix [8]	300	0.034	0.020	0.023	0.005	0.567	0.603	0.099	0.037	0.020	0.004	0.616	0.692
	400	0.043	0.025	0.026	0.005	0.560	0.590	0.146	0.066	0.070	0.018	0.602	0.665
	500	0.059	0.032	0.030	0.005	0.553	0.580	0.183	0.094	0.094	0.031	0.595	0.650
TTSA3R [30]	300	0.029	0.016	0.022	0.004	**0.567**	**0.603**	0.090	0.037	0.020	0.005	0.613	0.685
	400	0.035	0.019	0.023	0.004	0.561	0.592	0.104	0.045	0.035	0.005	0.607	0.673
	500	0.044	0.022	0.024	0.004	0.557	0.585	0.121	0.052	0.049	0.006	0.604	0.666
AFG-Pose (Ours)	300	0.025	0.012	0.021	**0.003**	0.566	0.600	0.075	0.026	**0.011**	0.004	**0.617**	0.690
	400	0.028	0.014	0.022	**0.003**	0.561	0.592	0.085	0.035	0.023	**0.003**	0.613	0.683
	500	0.031	0.015	0.022	**0.003**	0.558	0.587	0.101	0.040	0.033	**0.004**	0.611	0.680
AFG-Img (Ours)	300	**0.023**	**0.010**	**0.020**	**0.003**	0.563	0.596	**0.071**	**0.023**	**0.010**	**0.003**	0.616	**0.692**
	400	**0.023**	**0.010**	**0.020**	**0.003**	0.558	0.588	**0.075**	**0.026**	**0.015**	**0.003**	**0.617**	**0.689**
	500	**0.024**	**0.010**	**0.021**	**0.003**	0.554	0.581	**0.092**	**0.031**	**0.027**	**0.004**	**0.614**	**0.683**

A13 Qualitative 3D Reconstruction Comparison

Fig. 7 provides qualitative point-cloud reconstructions on representative 7-Scenes [17] and NRGBD [1] sequences, complementing the quantitative results in Table 18 (Appendix A12). Without adaptive gating, recurrent baselines accumulate pose drift and produce fragmented geometry; AFG variants yield more coherent surfaces and better-preserved scene structure across both datasets.

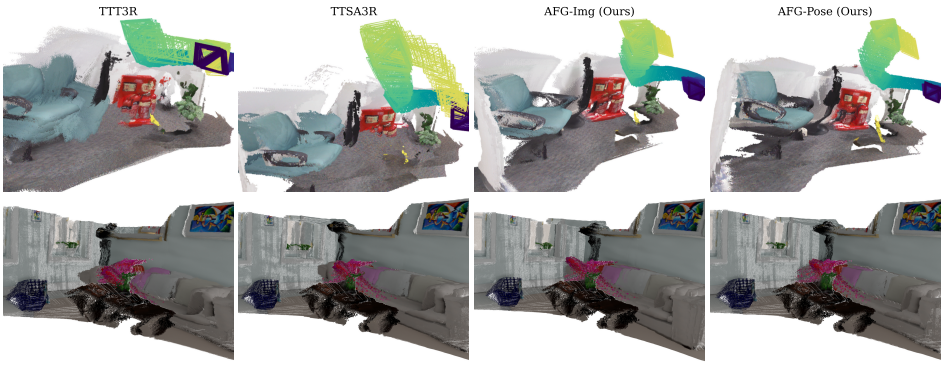


Figure 7: **Qualitative 3D reconstruction comparison on 7-Scenes (top) and NRGBD (bottom).** Without adaptive gating, recurrent baselines accumulate drift and produce fragmented geometry. Top: TTT3R and TTSA3R smear the armchair and the red fire extinguisher, while both AFG variants keep them intact. Bottom: TTT3R shows duplicated, smeared geometry around the flowers and the sofa, which AFG reduces.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes], [No], or [N/A].
- [N/A] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for [N/A]).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will also be asked to include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While [Yes] is generally preferable to [No], it is perfectly acceptable to answer [No] provided a proper justification is given (e.g., error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering [No] or [N/A] is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

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

Answer: [Yes]

Justification: The abstract and Sec. 1 state two claims: (i) TTT3R-style per-token gates fail on long sequences because structural averaging renders them frame-invariant, and (ii) Adaptive Frame Gating (AFG) restores frame-level adaptivity at inference time. Both are supported by the analysis in Sec. 3 and the multi-benchmark experiments in Sec. 5.

Guidelines:

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

2. **Limitations**

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

Answer: [Yes]

Justification: The Limitations and future work paragraph in Sec. 6 explicitly discusses (i) state-token saturation under sustained novel content as a failure mode AFG does not address, and (ii) correlation between the two variants' gating signals limiting naive frame-level fusion.

Guidelines:

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

3. **Theory assumptions and proofs**

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

Answer: [N/A]

Justification: The paper does not contain formal theorems. Sec. 3 provides an empirical and analytical diagnosis of the per-token gate's frame-invariance rather than formal theoretical results requiring proof.

Guidelines:

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

4. **Experimental result reproducibility**

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

Answer: [Yes]

Justification: The method is described in closed form in Sec. 4 and contains no learnable parameters or training procedure. Datasets, sequence-length sweeps, and alignment protocols are listed in Sec. 5; detailed per-length numerical values are tabulated in App. A8 and App. A11.

Guidelines:

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

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

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

   Answer: [Yes]

   Justification: Anonymized inference code will be released as supplementary material. The underlying base model (CUT3R) and all evaluation datasets (TUM-RGBD, ScanNet, Bonn, KITTI, 7-Scenes, NRGBD) are publicly available with standard academic-research licenses.

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: The proposed method is parameter-free and inference-only, so no training hyperparameters apply. Sec. 5 specifies datasets, sequence-length sweeps, alignment protocols (Sim(3) Umeyama for pose; metric and scale-and-shift for depth), and baseline configurations. The closed-form gate computation is given in Sec. 4.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [No]

   Justification: The proposed method is parameter-free and deterministic at inference, so per-sequence metrics carry no run-to-run variance. Across-sequence variability is implicit in the per-length tables of App. A8 and App. A11, where consistent trends across all sequence lengths support the main claims.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.

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

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: All experiments are run on a single NVIDIA A100 GPU with per-frame inference time and FPS reported in Sec. 5. AFG introduces no additional forward pass or trainable parameters beyond the base model, and the entire method requires no training compute.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: The work conforms with the NeurIPS Code of Ethics. We use publicly released benchmarks and a publicly available pretrained model for academic research; no human subjects are involved.

   Guidelines:

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

10. **Broader impacts**

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

Answer: [N/A]

Justification: The contribution is a methodological improvement to existing 3D reconstruction pipelines and does not introduce new applications or modalities; we identify no foreseeable specific positive or negative societal impacts beyond general improvements to visual odometry / SLAM accuracy.

Guidelines:

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

11. **Safeguards**

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

Answer: [N/A]

Justification: We do not release any high-risk pretrained model or scraped dataset. The contribution is an inference-time modification to a publicly available backbone with no elevated risk of misuse beyond the underlying model itself.

Guidelines:

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

12. **Licenses for existing assets**

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

Answer: [Yes]

Justification: All datasets (TUM-RGBD, ScanNet, Bonn, KITTI, 7-Scenes, NRGBD) and the CUT3R backbone are cited in the paper and used under their published academic-research licenses. Baseline methods (TTT3R, TTSA3R, MeMix, LongStream, Keyframe-VO) are cited.

Guidelines:

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

13. **New assets**

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

Answer: [N/A]

Justification: We release no new datasets or pretrained models. The algorithmic modification will be made available as anonymized code in the supplementary material with usage instructions.

Guidelines:

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

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

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

Answer: [N/A]

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

Guidelines:

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

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

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

    Answer: [N/A]

    Justification: The paper does not involve research with human subjects, so IRB approval is not applicable.

    Guidelines:

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

16. **Declaration of LLM usage**

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

    Answer: [N/A]

    Justification: LLMs were used only for writing assistance and do not constitute an original or core methodological component of the research.

    Guidelines:

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