HIST3R: Decomposing and Rectifying State Updates with History for Streaming 3D Reconstruction

Kenneth Yang[1] **Yu-Chiang Frank Wang**[1,2]
[1]National Taiwan University [2]NVIDIA
r13942093@ntu.edu.tw frankwang@nvidia.com

Abstract

Streaming 3D reconstruction with recurrent states enables long-video scene understanding under bounded memory, but turns reconstruction into a state-evolution problem: each incoming frame updates a compact scene memory on which all subsequent predictions depend, so unreliable updates overwrite historical context and accumulate into long-horizon geometric drift. We analyze CUT3R-style recurrent updates through a direction–magnitude decomposition, in which the magnitude sets the update gain, i.e., how strongly an observation is written into the state, and the direction encodes the geometric intent, i.e., where the update drives the state. This decomposition provides a unified view of CUT3R and its variants, revealing that existing training-free extensions perform magnitude-only control: they can attenuate an unreliable update but cannot correct its direction. We propose HIST3R, a training-free framework built on CUT3R that controls both components using history. HIST3R rectifies the update direction using locally consistent historical states, and modulates the update magnitude according to the visual discrepancy between the current observation and its history, suppressing redundant updates while preserving complementary evidence. Across camera pose estimation, 3D reconstruction, and video depth estimation, HIST3R consistently improves long-horizon performance while retaining the memory footprint and inference speed of CUT3R. With 1000 input frames, it reduces ATE by 23.87% on ScanNet and 33.65% on TUM-Dynamics relative to the strongest competing method; with 500 frames, it improves median reconstruction accuracy over the strongest CUT3R-based variant by 56.52% on 7-Scenes and 33.96% on NRGBD.

Project page:https://kennethyangg.github.io/HIST3R.

1 Introduction

Recovering dense 3D geometry from 2D observations is a central problem in computer vision, supporting applications across autonomous driving [1–3], augmented reality (AR) [4–7], and embodied AI [8–10]. Recent Transformer-based foundation models [11–13] have increasingly replaced classical optimization-based pipelines [14–18] due to their strong generalization ability and efficient feed-forward inference. Yet this advantage does not scale to long videos: reasoning jointly over an ever-growing set of views makes feed-forward reconstruction increasingly memory- and compute-intensive.

Streaming 3D reconstruction addresses this scalability issue by processing frames sequentially [19–27]. Existing methods preserve historical context through either **explicit memory** or **implicit memory**. Explicit-memory methods retain processed frames, features, or tokens in an external memory bank, preserving rich history at the cost of memory or computation that grows with sequence length [19–21]. Implicit-memory methods, including ours, instead compress history into a fixed-size recurrent state with bounded memory [23–25]. This compression, however, shifts the central

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

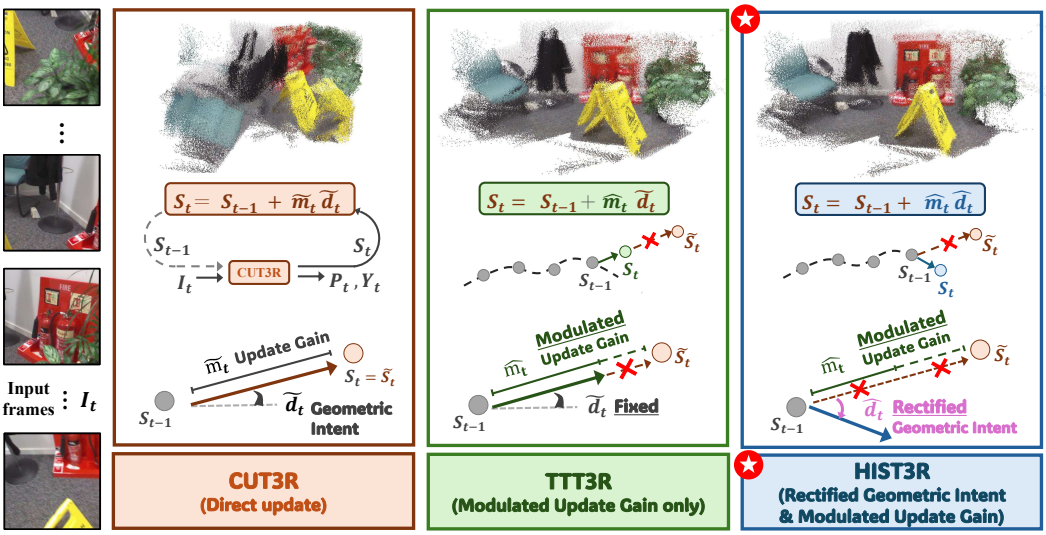


Figure 1: **A unified direction–magnitude view of recurrent state updates for long-sequence streaming 3D reconstruction.** Update magnitude and direction correspond to update gain and geometric intent, respectively. CUT3R [23] directly commits the candidate state, while magnitude-only CUT3R variants, represented by TTT3R [24], rescale the candidate update magnitude but keep its direction fixed. HIST3R controls both components by rectifying the update direction and modulating the update magnitude, enabling more reliable long-horizon state evolution.

challenge from per-frame reconstruction to recurrent state evolution: each incoming frame updates the only compact memory of the scene, so an unreliable update can overwrite useful history, and over many frames such errors accumulate into long-horizon geometric drift [24, 25].

We study this update problem through CUT3R [23], a representative implicit-memory streaming model. For each incoming frame, CUT3R produces a candidate update that writes the new observation into the recurrent state. Such an update must determine two things: how strongly the observation should modify the accumulated scene representation, and what geometric content it should contribute. We refer to the former as the **update gain** and the latter as the **geometric intent**. In the recurrent state space, these two roles correspond naturally to the magnitude and direction of the candidate update, which motivates a *direction–magnitude decomposition* of recurrent state updates.

As illustrated in Fig. 1, this decomposition provides a unified view of CUT3R and its recent training-free extensions. Vanilla CUT3R commits the candidate update as is, while TTT3R [24] and TTSA3R [25] can be interpreted as magnitude-only update control: they estimate a scalar gate from auxiliary signals, such as state-to-image attention or temporal–spatial dissimilarity, to modulate the update gain while leaving the update direction unchanged. As a result, they can attenuate an unreliable or redundant update, but cannot correct the geometric intent that determines where the recurrent state is driven. This limitation becomes increasingly critical over long input sequences. When an incoming frame induces a biased candidate update, magnitude-only gating may reduce its step size, yet the state still moves along the same biased direction. Since the recurrent state compactly carries past information, such directional biases persist and compound across frames, gradually pulling the state away from coherent scene structure and leading to geometric drift. Reliable long-horizon streaming reconstruction can therefore be viewed as a state-space evolution problem, requiring control over both how strongly the recurrent state is updated and how its geometric evolution is directed.

Motivated by this view, we propose **HIST3R**, a training-free framework built on CUT3R for long-sequence streaming 3D reconstruction. HIST3R decomposes the CUT3R candidate update into *direction* and *magnitude*, enabling separate control over geometric intent and update gain. To rectify geometric intent, HIST3R uses recent recurrent states as a local reference for plausible state evolution and adjusts the observation-induced update toward a history-consistent endpoint. To modulate the update gain, HIST3R estimates how much new evidence the incoming observation provides to the existing recurrent state. It therefore compares the current visual context with historical visual contexts

and adapts the update magnitude according to their discrepancy, suppressing redundant state changes while preserving responsiveness to observations that depart from recent history.

We evaluate HIST3R across camera pose estimation, 3D reconstruction, and video depth estimation under both standard and long-sequence streaming settings. Across these evaluations, HIST3R consistently improves long-horizon performance over prior streaming methods while maintaining competitive short-sequence accuracy, all without additional training. These results demonstrate that direction–magnitude decomposition provides an effective mechanism for controlling geometric intent and update gain in implicit-memory streaming reconstruction.

Our contributions are summarized as follows:

- We introduce a direction–magnitude decomposition of recurrent state updates, which unifies CUT3R and its variants and reveals update direction as the missing control axis for long-horizon reliability.
- We propose HIST3R, a training-free extension of CUT3R that rectifies geometric intent with historical recurrent states and modulates update gain with observation–history visual discrepancy, while preserving the memory footprint and inference speed of CUT3R.
- We demonstrate consistent long-horizon improvements across camera pose estimation, 3D reconstruction, and video depth estimation. At 1000 input frames, HIST3R reduces ATE over the second-best method, TTSA3R, by $\mathbf{23.87}\%$ on ScanNet and $\mathbf{33.65}\%$ on TUM-Dynamics; at 500 frames, it improves median reconstruction accuracy over TTSA3R by $\mathbf{56.52}\%$ on 7-Scenes and $\mathbf{33.96}\%$ on NRGBD.

2 Related Work

Offline Reconstruction Foundation Models. Recent learning-based methods have shifted 3D reconstruction from classical geometric pipelines [14–18] toward generalizable feed-forward foundation models [11–13, 28–31]. One line of work builds on pairwise reconstruction. DUSt3R [11] formulates 3D reconstruction as direct pointmap regression from image pairs, and subsequent methods improve matching, dynamic-scene robustness, and reconstruction quality within this pairwise paradigm [12, 28, 32]. Although effective for local two-view geometry, these methods still require global alignment or aggregation to reconstruct larger image collections.

Another line of work performs all-at-once multi-view reconstruction. VGGT [13] jointly processes multiple images with global transformer attention to predict camera poses, depth, pointmaps, and tracks, with recent variants improving inference speed, geometric priors, or scalability [33–37]. However, offline and all-at-once models assume a fixed input set and reason over all views jointly, making memory and computation grow rapidly as more frames are added. This scalability bottleneck motivates online reconstruction foundation models, which process frames incrementally while maintaining historical scene information.

Online Reconstruction Foundation Models. Online reconstruction foundation models can be broadly grouped into explicit-memory and implicit-memory approaches. Explicit-memory methods retain processed frames as externally accessible entries, such as spatial memories, cached features, or historical tokens. [19–22, 26, 27, 38]. Spann3R [38] adapts DUSt3R into an online setting by storing previous predictions as key-value features, while Point3R [19] introduces a spatial pointer memory associated with 3D positions. StreamVGGT [20] and STream3R [21] instead use causal attention with key-value caching to process incoming frames incrementally. Although these methods avoid full reprocessing of all frames, their memory or computation still tends to grow with sequence length. InfiniteVGGT [22] mitigates this issue through token pruning inspired by streaming language models [39, 40], but its pruning strategy mainly addresses cache efficiency rather than explicitly correcting accumulated geometric errors.

Implicit-memory methods, represented by CUT3R [23], compress history into a fixed-size recurrent state, keeping memory bounded regardless of sequence length [23–25] . Since all subsequent predictions depend on this compact state, reconstruction quality hinges on how each observation updates it. Recent training-free extensions improve update stability by rescaling the candidate update: TTT3R [24] views the recurrent update as test-time training and derives per-token learning rates from state-to-image attention, while TTSA3R [25] derives its gates from temporal–spatial

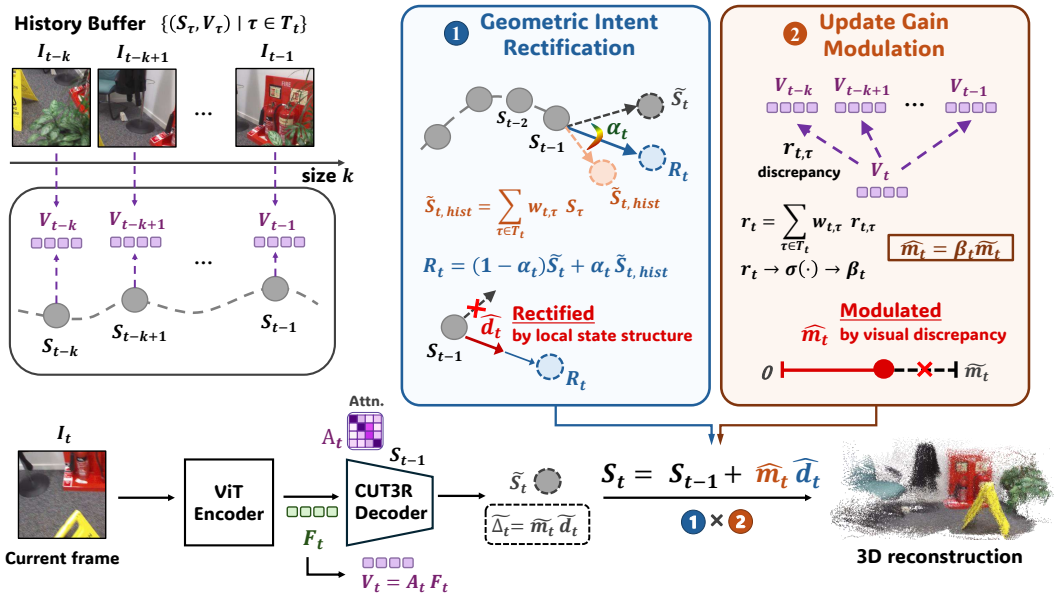


Figure 2: **Overview of HIST3R.** HIST3R maintains a sliding history buffer of past recurrent states and visual summaries. It decomposes the CUT3R candidate update into direction and magnitude to control geometric intent and update gain, respectively. Historical states rectify the update direction, while observation–history visual discrepancy modulates the update magnitude. The two components are combined into the final recurrent state update.

dissimilarity. Both regulate how strongly the candidate update is applied but leave its direction unchanged, corresponding to magnitude-only control under our decomposition. In contrast, HIST3R uses history to control both the direction and the magnitude of the update, while remaining training-free and preserving the bounded memory of implicit-memory methods.

3 Method

We present **HIST3R**, a training-free extension of CUT3R for long-sequence streaming 3D reconstruction. As illustrated in Fig. 2, HIST3R improves recurrent state updating by explicitly controlling the two factors introduced in Sec. 3.1: geometric intent and update gain.

3.1 Long-Horizon State Evolution and Update Decomposition

Recurrent state update in CUT3R. Given an online image stream $\{I_t\}_{t=1}^{T}$, streaming 3D reconstruction predicts the camera pose P_t and dense pointmap $Y_t \in \mathbb{R}^{H\times W\times 3}$ for each frame on the fly. CUT3R processes the stream with a persistent recurrent state $S_t \in \mathbb{R}^{N\times C}$, which compactly stores scene history using N state tokens. For each incoming frame, the image tokens $F_t \in \mathbb{R}^{P\times C}$ interact with the previous state S_{t-1} through interconnected transformer decoders:

$$F'_t, \tilde{S}_t = \text{Decoders}(F_t, S_{t-1}). \tag{1}$$

The decoders perform a dual interaction: image tokens read historical information from S_{t-1} to produce history-conditioned tokens F'_t, while state tokens integrate visual evidence from F_t to form the candidate state $\tilde{S}_t$. The resulting F'_t is fed into task-specific heads to predict P_t and Y_t.

Vanilla CUT3R commits the candidate state directly, $S_t = \tilde{S}_t$, which applies the candidate update $\tilde{\Delta}_t = \tilde{S}_t - S_{t-1}$ to the previous state without revision. Since S_t conditions all subsequent predictions, an unreliable candidate update, caused by factors such as motion blur, occlusion, or challenging viewpoints, is written directly into the compact scene memory and can weaken the historical context it encodes [24, 25].

Direction–magnitude decomposition. Each recurrent update plays two roles in reconstruction: it determines what geometric content the current observation contributes to the accumulated scene representation, which we call the *geometric intent*, and how strongly this content is committed, which we call the *update gain*. In recurrent state space, these two roles are naturally carried by the *direction* and *magnitude* of the candidate update, respectively. We therefore decompose the candidate update of each state token i as

$$\begin{aligned} \tilde{\Delta}_t^{(i)} &= \tilde{S}_t^{(i)} - S_{t-1}^{(i)} = \tilde{m}_t^{(i)} \tilde{d}_t^{(i)}, \\ \tilde{d}_t^{(i)} &= \frac{\tilde{\Delta}_t^{(i)}}{\tilde{m}_t^{(i)} + \epsilon}, \qquad \tilde{m}_t^{(i)} = \|\tilde{\Delta}_t^{(i)}\|_2, \end{aligned} \tag{2}$$

where $\tilde{d}_t^{(i)}$ and $\tilde{m}_t^{(i)}$ denote the candidate geometric intent and candidate update gain, respectively.

This decomposition provides a unified update-control view of CUT3R and its variants. Vanilla CUT3R commits both the candidate geometric intent and update gain without revision. TTT3R and TTSA3R [24, 25] can be interpreted as *magnitude-only* modulation: they estimate a scalar gate $g_t^{(i)} \in [0, 1]$ from auxiliary signals, such as attention-based confidence or temporal–spatial dissimilarity, and use it to modulate the update gain while keeping the geometric intent fixed:

$$S_t^{(i)} = S_{t-1}^{(i)} + g_t^{(i)} \tilde{m}_t^{(i)} \tilde{d}_t^{(i)}. \tag{3}$$

Such gating can attenuate an unreliable update, but cannot correct where the state is driven: a biased geometric intent continues to drive the state, only with a smaller step. Over long sequences, such uncorrected geometric intent accumulates and gradually moves the recurrent state away from coherent evolution.

HIST3R addresses this missing control axis by controlling both components with history. It rectifies the geometric intent through historical state consistency to obtain $\hat{d}_t^{(i)}$, and modulates the update gain through observation–history visual discrepancy to obtain $\hat{m}_t^{(i)}$. The final recurrent update is

$$S_t^{(i)} = S_{t-1}^{(i)} + \hat{m}_t^{(i)} \hat{d}_t^{(i)}. \tag{4}$$

Sec. 3.2 and Sec. 3.3 describe how $\hat{d}_t^{(i)}$ and $\hat{m}_t^{(i)}$ are computed, respectively.

3.2 Rectifying Geometric Intent via Historical State Consistency

The decomposition in Sec. 3.1 identifies the geometric intent as the component that determines where the recurrent state evolves. A reliable geometric intent should therefore not be determined solely by the current frame, but should also respect how the scene representation has evolved in recent history. As shown in Appendix A.1, CUT3R states support using recent history as such a local reference: adjacent states have high cosine affinity, and their variations are largely captured by a low-dimensional subspace. Under this local structure, a compatible candidate endpoint should be explainable by nearby historical states, following the self-expressiveness principle in subspace modeling [41] and classical geometric priors based on local low-dimensional subspaces [42–44].

Based on this observation, HIST3R rectifies the geometric intent by correcting the endpoint that defines it. For each state token i, HIST3R blends the CUT3R candidate endpoint $\tilde{S}_t^{(i)}$ with a history-proposed endpoint $\tilde{S}_{t,\mathrm{hist}}^{(i)}$, and derives the rectified direction from the resulting endpoint $R_t^{(i)}$:

$$R_t^{(i)} = (1 - \alpha_t^{(i)}) \tilde{S}_t^{(i)} + \alpha_t^{(i)} \tilde{S}_{t,\mathrm{hist}}^{(i)}, \qquad \hat{d}_t^{(i)} = \frac{R_t^{(i)} - S_{t-1}^{(i)}}{\left\| R_t^{(i)} - S_{t-1}^{(i)} \right\|_2 + \epsilon}, \tag{5}$$

where $\alpha_t^{(i)} \in [0, 1]$ is an alignment gate that determines how much history should guide the update with $\alpha_t^{(i)} = 0$ recovering the candidate geometric intent of CUT3R. We next describe how $\tilde{S}_{t,\mathrm{hist}}^{(i)}$ and $\alpha_t^{(i)}$ are constructed.

History-proposed endpoint. HIST3R maintains a sliding history buffer of the most recent k recurrent states and visual summaries:

$$\mathcal{H}_t = \{(S_\tau, V_\tau) \mid \tau \in \mathcal{T}_t\}, \qquad \mathcal{T}_t = \{\max(1, t-k), \ldots, t-1\}. \tag{6}$$

The cached states $\{S_\tau\}$ are used here, while the visual summaries $\{V_\tau\}$ are used in Sec. 3.3 to modulate the update gain. To construct the history-proposed endpoint, HIST3R retrieves locally similar past states in a token-wise manner. It computes cosine affinities $a_{t,\tau}^{(i)} = \cos(\tilde{S}_t^{(i)}, S_\tau^{(i)})$ for $\tau \in \mathcal{T}_t$, normalizes them into local history weights, and aggregates the historical states accordingly:

$$w_{t,\tau}^{(i)} = \frac{\exp(a_{t,\tau}^{(i)})}{\sum_{\tau' \in \mathcal{T}_t} \exp(a_{t,\tau'}^{(i)})}, \qquad \tilde{S}_{t,\text{hist}}^{(i)} = \sum_{\tau \in \mathcal{T}_t} w_{t,\tau}^{(i)} S_\tau^{(i)}. \tag{7}$$

The resulting $\tilde{S}_{t,\text{hist}}^{(i)}$ expresses the candidate endpoint through nearby historical states, reflecting the structure of recent state evolution.

Alignment-aware rectification. The history-proposed endpoint should guide the update only when it agrees with the current candidate. High alignment indicates that recent states provide a reliable reference for rectifying the candidate geometric intent, while low alignment suggests that the current observation introduces a genuine change and should not be over-constrained by history. We therefore define the alignment gate as

$$c_t^{(i)} = \cos\Big(\tilde{S}_t^{(i)}, \tilde{S}_{t,\text{hist}}^{(i)}\Big), \qquad \alpha_t^{(i)} = \sigma\Big(\lambda_{\text{align}}\Big(c_t^{(i)} - \theta_{\text{align}}\Big)\Big), \tag{8}$$

where λ_{align} and θ_{align} control the sharpness and threshold of the gate. The resulting $\hat{d}_t^{(i)}$ in Eq. (5) serves as the rectified geometric intent of the HIST3R update, and Sec. 3.3 next determines the update gain along it.

3.3 Modulating Update Gain via Observation Discrepancy

After rectifying the geometric intent, HIST3R modulates the update gain, i.e., how strongly the current observation should modify the recurrent state along $\hat{d}_t^{(i)}$. A suitable update gain depends on how much new evidence the observation provides relative to recent history: if the observation is already consistent with recent visual history, a large update may redundantly perturb the recurrent state; if it provides complementary geometric evidence not captured by recent history, an overly conservative update may prevent the state from adapting.

HIST3R therefore modulates the candidate update gain according to an observation discrepancy score $r_t^{(i)}$, which measures how much the current visual context differs from its historical context:

$$\beta_t^{(i)} = \sigma\Big(\lambda_{\text{disc}}\Big(r_t^{(i)} - \theta_{\text{disc}}\Big)\Big), \qquad \hat{m}_t^{(i)} = \beta_t^{(i)} \tilde{m}_t^{(i)}, \tag{9}$$

where λ_{disc} and θ_{disc} control the sharpness and threshold of the gate. A small discrepancy indicates that the observation is largely explained by recent history, yielding a conservative update gain that suppresses redundant state changes, while a large discrepancy indicates new evidence and allows a stronger update along $\hat{d}_t^{(i)}$. Combined with $\hat{d}_t^{(i)}$ from Sec. 3.2, $\hat{m}_t^{(i)}$ yields the final HIST3R update in Eq. 4. We next describe how the visual context is summarized and how $r_t^{(i)}$ is computed.

State-token-specific visual summary. Computing $r_t^{(i)}$ requires a compact summary of each frame that can be compared across time. A simple choice is to average-pool all image tokens into a single global token. However, different state tokens attend to different image regions when interacting with the frame through the decoders in Eq. 1, so a single global summary discards the visual evidence relevant to each state token. HIST3R instead summarizes each frame per state token, aggregating image tokens according to the first-layer state-to-image attention map $A_t \in \mathbb{R}^{N \times P}$ from the decoder:

$$V_t^{(i)} = \sum_{j=1}^{P} A_t^{(i,j)} F_t^{(j)}, \tag{10}$$

where $F_t^{(j)}$ denotes the j-th image token. Each $V_t^{(i)}$ thus represents the visual evidence observed by state token i. The resulting $V_t \in \mathbb{R}^{N \times C}$ is stored in the history buffer $\mathcal{H}_t$ defined in Eq. 6.

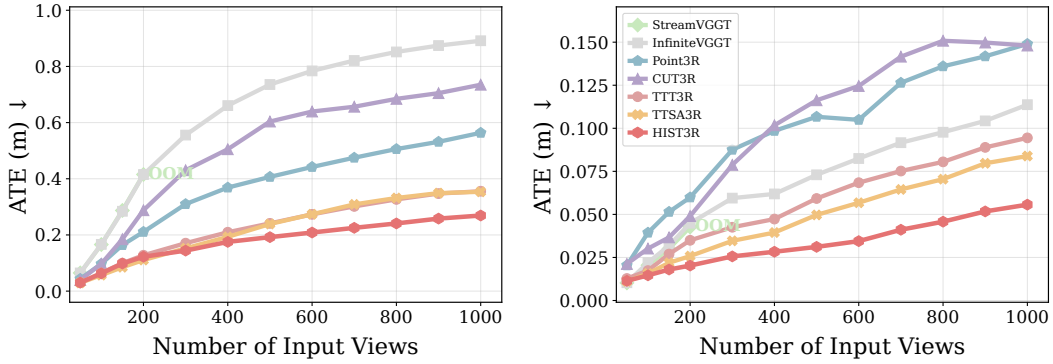


Figure 3: **Camera pose estimation on ScanNet [45] (left) and TUM-Dynamics [46] (right).** OOM denotes out-of-memory under an 80GB GPU memory budget.

Discrepancy-guided modulation. The discrepancy score $r_t^{(i)}$ measures how much additional visual evidence the current observation provides beyond relevant history. For each state token i, HIST3R compares the current summary $V_t^{(i)}$ with each historical summary $V_\tau^{(i)}$ using a normalized L2 distance, and aggregates these distances with the local history weights $w_{t,\tau}^{(i)}$ from Eq. 7:

$$r_{t,\tau}^{(i)} = \frac{\left\| V_t^{(i)} - V_\tau^{(i)} \right\|_2}{\left\| V_\tau^{(i)} \right\|_2 + \epsilon}, \qquad r_t^{(i)} = \sum_{\tau \in \mathcal{T}_t} w_{t,\tau}^{(i)} r_{t,\tau}^{(i)}. \tag{11}$$

Since $w_{t,\tau}^{(i)}$ assigns larger weights to historical states that are more similar to the current candidate state, the discrepancy is measured mainly against the most relevant history rather than against the entire history uniformly. The resulting $r_t^{(i)}$ determines the update gain through Eq. 9.

Together with the rectified geometric intent $\hat{d}_t^{(i)}$ from Sec. 3.2, this update gain yields the final HIST3R update in Eq. 4, which controls both components of the recurrent update using history while remaining training-free.

4 Experiments

We evaluate HIST3R across camera pose estimation (Sec. 4.1), 3D reconstruction (Sec. 4.2), and video depth estimation (Sec. 4.3), followed by ablations that validate the proposed direction–magnitude decomposition view (Sec. 4.4).

Baselines and protocol. We compare HIST3R with recent online reconstruction methods. CUT3R [23] serves as the base implicit-memory model. Our main comparisons are TTT3R [24] and TTSA3R [25], two training-free CUT3R extensions that modulate the update gain while leaving the geometric intent unchanged. We also compare with explicit-memory methods, including Point3R [19], StreamVGGT [20], and InfiniteVGGT [22]. All hyperparameters are selected once on the ScanNet training split and remain fixed across all datasets, tasks, and sequence lengths, as detailed in Appendix A.2. Since HIST3R is designed for long-horizon streaming reconstruction, the main paper focuses on long-sequence evaluation. Sparse-view and standard short-sequence results are reported in Appendix B. All methods are evaluated on a single NVIDIA H100 GPU with 80GB memory.

4.1 Camera Pose Estimation

For camera pose estimation, we evaluate on ScanNet [45] and TUM-Dynamics [46], following the protocol of prior online reconstruction methods [23, 24]. We use Absolute Trajectory Error (ATE) after Sim(3) trajectory alignment [46] as the main evaluation metric.

As shown in Fig. 3, HIST3R consistently achieves the lowest ATE across long input sequences on both ScanNet and TUM-Dynamics. The improvement becomes more pronounced as the sequence

Table 1: **3D reconstruction results on 7-Scenes [47] and NRGBD [48]**. Best results are shown in bold, and second-best results are underlined.

Method	Input	7-Scenes						NRGBD					
		Acc. ↓		Comp. ↓		NC ↑		Acc. ↓		Comp. ↓		NC ↑	
		Mean	Med.	Mean	Med.	Mean	Med.	Mean	Med.	Mean	Med.	Mean	Med.
StreamVGGT	*300*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*
InfiniteVGGT		0.042	0.017	0.026	0.007	**0.570**	**0.607**	**0.054**	0.031	0.025	0.005	**0.645**	**0.748**
Point3R		0.046	0.025	0.033	0.014	0.564	0.597	0.073	0.040	0.013	**0.004**	0.614	0.687
CUT3R		0.126	0.085	0.070	0.030	0.543	0.564	0.215	0.118	0.064	0.013	0.570	0.607
TTT3R		0.041	0.025	0.024	0.005	0.565	0.599	0.101	0.040	0.022	**0.004**	0.607	0.675
TTSA3R		0.029	0.015	0.022	**0.004**	0.567	0.603	0.090	0.037	0.020	0.005	0.613	0.684
HIST3R		**0.020**	**0.009**	**0.021**	**0.004**	0.565	0.599	0.075	**0.025**	**0.012**	**0.004**	0.620	0.695
StreamVGGT	*400*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*
InfiniteVGGT		0.040	0.015	0.027	0.006	**0.565**	**0.599**	**0.066**	0.035	0.035	0.007	**0.647**	**0.752**
Point3R		0.053	0.026	0.025	0.009	0.560	0.591	0.090	0.043	**0.024**	**0.004**	0.610	0.681
CUT3R		0.152	0.102	0.094	0.041	0.536	0.552	0.305	0.220	0.108	0.038	0.554	0.580
TTT3R		0.052	0.031	0.027	0.005	0.557	0.586	0.142	0.066	0.053	0.008	0.593	0.647
TTSA3R		0.035	0.019	**0.023**	**0.004**	0.560	0.591	0.103	0.046	0.035	0.006	0.608	0.674
HIST3R		**0.021**	**0.010**	**0.023**	**0.004**	0.561	0.592	0.084	**0.034**	**0.024**	**0.004**	0.616	0.688
StreamVGGT	*500*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*
InfiniteVGGT		0.038	0.014	0.023	0.005	**0.562**	**0.593**	**0.070**	0.046	0.037	0.008	**0.641**	**0.741**
Point3R		0.054	0.026	0.025	0.010	0.555	0.583	0.114	0.049	**0.026**	**0.004**	0.611	0.683
CUT3R		0.179	0.126	0.093	0.028	0.530	0.544	0.329	0.244	0.151	0.050	0.549	0.571
TTT3R		0.066	0.039	0.029	0.006	0.550	0.575	0.166	0.091	0.065	0.008	0.584	0.633
TTSA3R		0.045	0.023	0.025	**0.004**	0.555	0.583	0.120	0.053	0.050	0.006	0.604	0.667
HIST3R		**0.023**	**0.010**	**0.022**	**0.004**	0.557	0.586	0.092	**0.035**	0.033	**0.004**	0.610	0.678

length increases: at 1000 input frames, HIST3R reduces ATE over the second-best method, TTSA3R, by $\mathbf{23.87}\%$ on ScanNet and $\mathbf{33.65}\%$ on TUM-Dynamics. This trend supports our motivation that long-horizon streaming reconstruction benefits from controlling not only the update gain but also the geometric intent.

4.2 3D Reconstruction

Following prior work [19, 22, 23], we evaluate 3D reconstruction on 7-Scenes [47] and NRGBD [48]. We assess reconstruction quality by comparing predicted pointmaps with ground-truth point clouds using accuracy, completeness, and normal consistency (NC). Accuracy measures the nearest-neighbor distance from the predicted reconstruction to the ground truth, while completeness measures the distance in the reverse direction; for both metrics, we report mean and median values. To evaluate long-sequence reconstruction, we test each method on input sequences of 300, 400, and 500 frames.

As shown in Table 1, HIST3R achieves the strongest reconstruction quality on 7-Scenes, obtaining the best accuracy and completeness across long input sequences while remaining competitive in normal consistency. On NRGBD, HIST3R achieves the best or second-best results on most metrics. Compared with explicit-memory methods such as InfiniteVGGT and Point3R, no single method dominates every metric, and HIST3R remains competitive in accuracy and completeness. Its main advantage lies in efficiency: by maintaining a compact recurrent state, HIST3R requires substantially less memory and runs considerably faster than explicit-memory methods, as analyzed in Appendix A.4.

The advantage is especially clear within the CUT3R-based implicit-memory family. HIST3R consistently outperforms CUT3R, TTT3R, and TTSA3R across nearly all datasets, sequence lengths, and reconstruction metrics. At 500 frames, HIST3R improves median accuracy over TTSA3R, the second-best CUT3R variant, by $\mathbf{56.52}\%$ on 7-Scenes and $\mathbf{33.96}\%$ on NRGBD. This margin indicates that controlling both geometric intent and update gain substantially strengthens CUT3R's long-horizon reconstruction capability.

4.3 Video Depth Estimation

Following prior online reconstruction work [23, 24], we evaluate video depth estimation on KITTI [50] and Bonn [49], which cover outdoor driving and dynamic indoor scenes. We report absolute relative error (AbsRel) and $\delta < 1.25$, the percentage of predicted depths within a factor of 1.25 of the ground truth, under two protocols: scale-aligned evaluation, which aligns each predicted

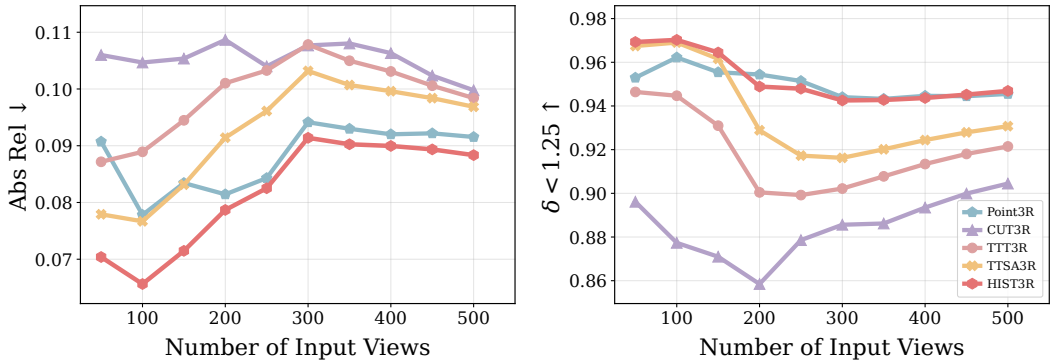


Figure 4: **Metric depth evaluation on Bonn [49].** We exclude VGGT-based methods that do not support metric depth prediction.

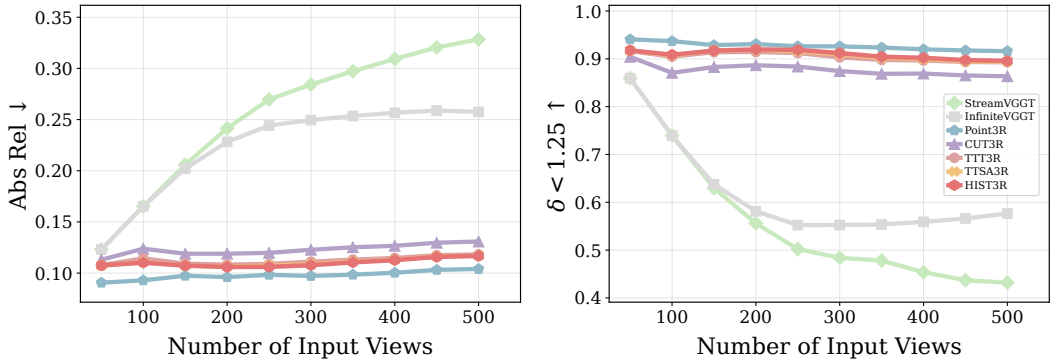


Figure 5: **Scale-aligned depth evaluation on KITTI [50].**

sequence to the ground truth with a per-sequence scale to measure relative depth quality, and metric-depth evaluation, which compares predictions in metric units without alignment and is reported only for methods that output metric-scale pointmaps.

As shown in Fig. 4, HIST3R achieves the strongest metric-depth performance on Bonn across both AbsRel and $\delta < 1.25$, especially under longer input sequences. On KITTI, HIST3R ranks second behind Point3R in scale-aligned depth evaluation and remains competitive, as shown in Fig. 5.

4.4 Ablation Studies

We ablate the two update controls introduced by the direction–magnitude decomposition, as well as the design choices within each.

Table 2: **Ablation of direction–magnitude decomposition.** We evaluate direction rectification, magnitude modulation, and their combination on camera pose estimation, scale-aligned video depth estimation, and 3D reconstruction, all under the 300-frame setting. AR denotes AbsRel, δ denotes $\delta < 1.25$, and reconstruction Acc./Comp. are reported as mean accuracy/completeness.

Method	Update Rule	TUM-D	Bonn		KITTI		7-Scenes		NRGBD	
		ATE ↓	AR ↓	δ ↑	AR ↓	δ ↑	Acc. ↓	Comp. ↓	Acc. ↓	Comp. ↓
CUT3R baseline	$S_t = S_{t-1} + \widetilde{\mathbf{m}}_t \widetilde{\mathbf{d}}_t$	0.079	0.089	0.938	0.123	0.874	0.126	0.070	0.215	0.064
+ Direction rect.	$S_t = S_{t-1} + \widetilde{\mathbf{m}}_t \widehat{\mathbf{d}}_t$	0.032	**0.077**	0.952	0.109	0.907	0.027	0.023	0.086	0.017
+ Magnitude mod.	$S_t = S_{t-1} + \widehat{\mathbf{m}}_t \widetilde{\mathbf{d}}_t$	0.027	0.080	0.952	0.109	0.909	0.021	**0.021**	0.077	**0.012**
HIST3R	$S_t = S_{t-1} + \widehat{\mathbf{m}}_t \widehat{\mathbf{d}}_t$	**0.026**	**0.077**	**0.954**	**0.108**	**0.912**	**0.020**	**0.021**	**0.075**	**0.012**

Table 3: **Design ablations** of direction rectification (left) and magnitude modulation (right), evaluated under the 200-frame setting with scale-aligned depth. The check marks indicate whether cosine-weighted aggregation, the alignment gate α_t, and direction-only rectification are applied.

Direction Rectification: $S_t = S_{t-1} + \tilde{\mathbf{m}}_t \tilde{\mathbf{d}}_t$

Design	Cos. Agg.	Align. Gate	Dir. Only	TUM ATE ↓	KITTI AR ↓	KITTI δ ↑
Protocol ($k = 8$)	✓	✓	✓	**0.024**	**0.107**	**0.917**
	✗	✓	✓	0.026	0.110	0.895
	✓	✗	✓	0.037	0.119	0.871
	✓	✓	✗	0.028	0.123	0.863
Window size k		$k = 2$		0.044	0.114	0.896
		$k = 4$		0.035	0.110	0.908
		$k = 8$ *(default)*		0.024	**0.107**	**0.917**
		$k = 16$		**0.021**	0.108	0.916
		$k = 32$		**0.021**	0.108	0.916
		$k = 64$		0.023	0.110	0.913

Magnitude Modulation: $S_t = S_{t-1} + \tilde{\mathbf{m}}_t \tilde{\mathbf{d}}_t$

Design	Attn. Layers	TUM ATE ↓	KITTI AR ↓	KITTI δ ↑
Avg. pooling	–	0.043	0.112	0.903
Attn. pooling	$n = 1$ *(default)*	**0.021**	**0.106**	0.918
	$n = 3$	0.023	**0.106**	0.919
	$n = 6$	0.023	**0.106**	**0.920**
	$n = 12$	0.023	**0.106**	0.919

Component ablation. We isolate the two update controls by applying only direction rectification or only magnitude modulation. As shown in Table 2, both improve over CUT3R across camera pose estimation, video depth estimation, and 3D reconstruction, and their combination achieves the best overall performance. This supports the decomposition view in Eq. 4: reliable long-horizon updating benefits from jointly rectifying the geometric intent and modulating the update gain.

Design ablations. Table 3 ablates the design choices within each component, applying direction rectification and magnitude modulation separately. Direction rectification consists of three stages: aggregating historical states with cosine weights to form the history-proposed endpoint in Eq. 7, fusing it with the candidate endpoint through the alignment gate in Eq. 8, and using the rectified endpoint only to set the update direction in Eq. 5. Removing any of these stages degrades performance. Replacing cosine weights with uniform averaging weakens the history proposal, and removing the alignment gate, which imposes history unconditionally, causes the largest drop. Directly adopting the rectified endpoint as the new state, which alters both direction and magnitude, also degrades performance, supporting the separate control of the two components. For the history window, performance improves as k increases and remains stable for $8 \leq k \leq 32$, while an overly long window introduces less relevant states; we therefore use $k = 8$ as the smallest stable window.

For magnitude modulation, state-token-specific attention pooling clearly outperforms global average pooling, confirming that different state tokens rely on different image regions. Using only the first-layer attention map performs on par with aggregating multiple layers while allowing the remaining layers to keep FlashAttention [51], which keeps HIST3R nearly as fast as CUT3R, as shown in Appendix A.4.

5 Conclusion

We presented HIST3R, a training-free framework that improves long-sequence streaming 3D reconstruction on top of CUT3R. The key idea is to view recurrent state updates as controllable state-space evolution and to decompose each update into direction and magnitude, which correspond to its geometric intent and update gain. This decomposition unifies CUT3R and its training-free variants, and reveals that existing variants control only the update gain, leaving the geometric intent as a missing control axis. HIST3R controls both components with history: it rectifies the geometric intent using the local structure of recent recurrent states, and modulates the update gain using observation–history visual discrepancy. Together, these two controls allow the compact recurrent state to incorporate useful new evidence while avoiding redundant or unstable updates over long sequences. Experiments across camera pose estimation, 3D reconstruction, and video depth estimation show that HIST3R consistently improves long-horizon performance while preserving the memory and speed of CUT3R. Overall, our results suggest that controlling both where and how strongly the recurrent state is updated provides an effective principle for robust long-horizon 3D perception.

Acknowledgments and Disclosure of Funding

This work is supported in part by the National Science and Technology Council (NSTC), Taiwan, under grants NSTC 114-2221-E-002-056-MY2 and NSTC 114-2634-F-002-006. We also thank the National Center for High-performance Computing (NCHC) for providing computational and storage resources.

References

[1] Yi Wei, Linqing Zhao, Wenzhao Zheng, Zheng Zhu, Jie Zhou, and Jiwen Lu. SurroundOcc: Multi-camera 3D occupancy prediction for autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 21672–21683, 2023.

[2] Yuanhui Huang, Wenzhao Zheng, Yunpeng Zhang, Jie Zhou, and Jiwen Lu. Tri-perspective view for vision-based 3D semantic occupancy prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9223–9232, 2023.

[3] Xiaoyu Tian, Tao Jiang, Longfei Yun, Yucheng Mao, Huitong Yang, Yue Wang, Yilun Wang, and Hang Zhao. Occ3D: A large-scale 3D occupancy prediction benchmark for autonomous driving. In *Advances in Neural Information Processing Systems*, pages 64318–64330, 2023.

[4] Richard A. Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J. Davison, Pushmeet Kohli, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. KinectFusion: Real-time dense surface mapping and tracking. In *IEEE International Symposium on Mixed and Augmented Reality*, pages 127–136, 2011.

[5] Jin Gyu Hong, Seung Young Noh, Hee Kyung Lee, Won Sik Cheong, and Ju Yong Chang. 3D clothed human reconstruction from sparse multi-view images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 677–687, 2024.

[6] Shunyuan Zheng, Boyao Zhou, Ruizhi Shao, Boning Liu, Shengping Zhang, Liqiang Nie, and Yebin Liu. GPS-Gaussian: Generalizable pixel-wise 3D Gaussian splatting for real-time human novel view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19680–19690, 2024.

[7] Jiahui Lei, Yijia Weng, Adam W. Harley, Leonidas Guibas, and Kostas Daniilidis. MoSca: Dynamic Gaussian fusion from casual videos via 4D motion scaffolds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6165–6177, 2025.

[8] Yuqi Wu, Wenzhao Zheng, Sicheng Zuo, Yuanhui Huang, Jie Zhou, and Jiwen Lu. EmbodiedOcc: Embodied 3D occupancy prediction for vision-based online scene understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 26360–26370, 2025.

[9] Muzhi Han, Zeyu Zhang, Ziyuan Jiao, Xu Xie, Yixin Zhu, Song-Chun Zhu, and Hangxin Liu. Scene reconstruction with functional objects for robot autonomy. *International Journal of Computer Vision*, 130(12):2940–2961, 2022.

[10] Zike Yan, Haoxiang Yang, and Hongbin Zha. Active neural mapping. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10947–10958, 2023.

[11] 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.

[12] Vincent Leroy, Yohann Cabon, and Jerome Revaud. Grounding image matching in 3D with MASt3R. In *European Conference on Computer Vision*, pages 71–91, 2024.

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

[14] Noah Snavely, Steven M. Seitz, and Richard Szeliski. Photo tourism: Exploring photo collections in 3D. *ACM Transactions on Graphics*, 25(3):835–846, 2006.

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

[16] Steven M. Seitz, Brian Curless, James Diebel, Daniel Scharstein, and Richard Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 519–528, 2006.

[17] Yasutaka Furukawa and Jean Ponce. Accurate, dense, and robust multi-view stereopsis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(8):1362–1376, 2010.

[18] Johannes L. Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision*, pages 501–518, 2016.

[19] Yuqi Wu, Wenzhao Zheng, Jie Zhou, and Jiwen Lu. Point3R: Streaming 3D reconstruction with explicit spatial pointer memory. In *Advances in Neural Information Processing Systems*, pages 69675–69699, 2025.

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

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

[22] 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.

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

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

[25] 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.

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

[27] Zhisong Xu and Takeshi Oishi. FrameVGGT: Coherence-preserving memory for bounded streaming geometry. *arXiv preprint arXiv:2603.07690*, 2026.

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

[29] Jianing Yang, Alexander Sax, Kevin J. Liang, Mikael Henaff, Hao Tang, Ang Cao, Joyce Chai, Franziska Meier, and Matt Feiszli. Fast3R: Towards 3D reconstruction of 1000+ images in one forward pass. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21924–21935, 2025.

[30] Zhenggang Tang, Yuchen Fan, Dilin Wang, Hongyu Xu, Rakesh Ranjan, Alexander Schwing, and Zhicheng Yan. MV-DUSt3R+: Single-stage scene reconstruction from sparse views in 2 seconds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5283–5293, 2025.

[31] Shangzhan Zhang, Jianyuan Wang, Yinghao Xu, Nan Xue, Christian Rupprecht, Xiaowei Zhou, Yujun Shen, and Gordon Wetzstein. FLARE: Feed-forward geometry, appearance and camera estimation from uncalibrated sparse views. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21936–21947, 2025.

[32] Xingyu Chen, Yue Chen, Yuliang Xiu, Andreas Geiger, and Anpei Chen. Easi3R: Estimating disentangled motion from DUSt3R without training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9158–9168, 2025.

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

[34] Yifan Wang, Jianjun Zhou, Haoyi Zhu, Wenzheng Chang, Yang Zhou, Zizun Li, Junyi Chen, Jiangmiao Pang, Chunhua Shen, and Tong He. π^3: Permutation-equivariant visual geometry learning. In *International Conference on Learning Representations*, 2026.

[35] Haotong Lin, Sili Chen, Jun Hao Liew, Donny Y. Chen, Zhenyu Li, Yang Zhao, Sida Peng, Hengkai Guo, Xiaowei Zhou, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth anything 3: Recovering the visual space from any views. In *International Conference on Learning Representations*, 2026.

[36] Xinze Li, Pengxu Chen, Yiyuan Wang, Weifeng Su, and Wentao Cheng. S-VGGT: Structure-aware subscene decomposition for scalable 3D foundation models. In *IEEE International Conference on Multimedia and Expo*, 2026.

[37] Zipeng Wang and Dan Xu. FlashVGGT: Efficient and scalable visual geometry transformers with compressed descriptor attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21826–21835, 2026.

[38] Hengyi Wang and Lourdes Agapito. 3D reconstruction with spatial memory. In *International Conference on 3D Vision*, pages 78–89, 2025.

[39] Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. In *International Conference on Learning Representations*, 2024.

[40] Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, Zhangyang Wang, and Beidi Chen. H2O: Heavy-hitter oracle for efficient generative inference of large language models. In *Advances in Neural Information Processing Systems*, pages 34661–34710, 2023.

[41] Ehsan Elhamifar and René Vidal. Sparse subspace clustering: Algorithm, theory, and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(11):2765–2781, 2013.

[42] Yingying Zhu, Dong Huang, Fernando De La Torre, and Simon Lucey. Complex non-rigid motion 3D reconstruction by union of subspaces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1542–1549, 2014.

[43] Suryansh Kumar, Yuchao Dai, and Hongdong Li. Spatio-temporal union of subspaces for multi-body non-rigid structure-from-motion. *Pattern Recognition*, 71:428–443, 2017.

[44] Suryansh Kumar, Anoop Cherian, Yuchao Dai, and Hongdong Li. Scalable dense non-rigid structure-from-motion: A Grassmannian perspective. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 254–263, 2018.

[45] 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 2432–2443, 2017.

[46] Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. A benchmark for the evaluation of RGB-D SLAM systems. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 573–580, 2012.

[47] 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.

[48] 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.

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

[50] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The KITTI dataset. *The International Journal of Robotics Research*, pages 1231–1237, 2013.

[51] Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. FlashAttention: Fast and memory-efficient exact attention with IO-Awareness. *Advances in Neural Information Processing Systems*, 35:16344–16359, 2022.

[52] Haoyi Zhu, Yifan Wang, Jianjun Zhou, Wenzheng Chang, Yang Zhou, Zizun Li, Junyi Chen, Chunhua Shen, Jiangmiao Pang, and Tong He. Aether: Geometric-aware unified world modeling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8535–8546, 2025.

[53] Daniel J Butler, Jonas Wulff, Garrett B Stanley, and Michael J Black. A naturalistic open source movie for optical flow evaluation. In *European Conference on Computer Vision*, pages 611–625, 2012.

[54] Johannes Kopf, Xuejian Rong, and Jia-Bin Huang. Robust consistent video depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1611–1621, 2021.

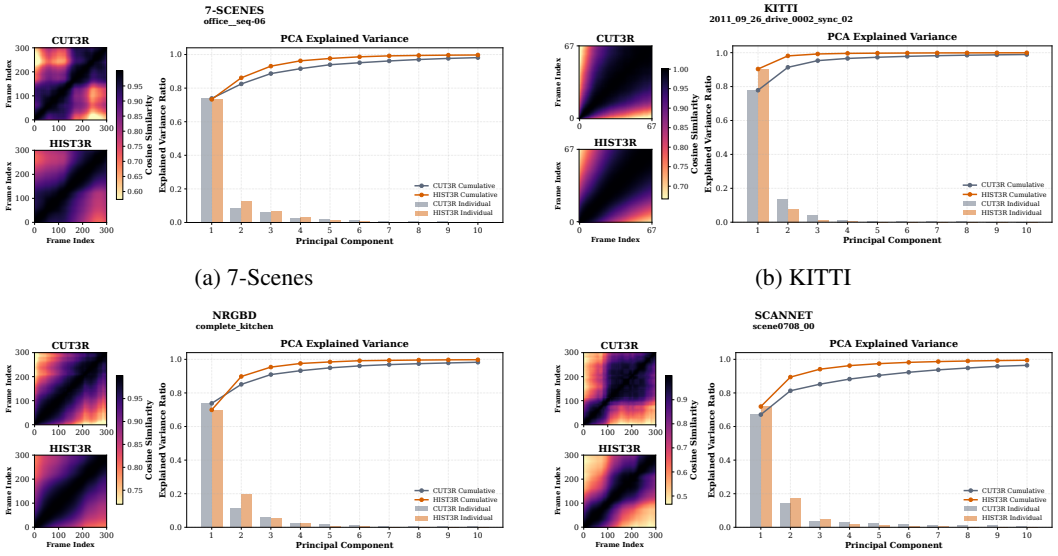
[55] Zhoutong Zhang, Forrester Cole, Zhengqi Li, Michael Rubinstein, Noah Snavely, and William T Freeman. Structure and motion from casual videos. In *European Conference on Computer Vision*, pages 20–37. Springer, 2022.

Appendix

A More Experimental Analysis

We provide additional analyses and results in this section. We first analyze the local structure of CUT3R state evolution in Appendix A.1, which motivates our history-guided direction rectification. We then detail the hyperparameter selection protocol and sensitivity of HIST3R in Appendix A.2, and show in Appendix A.3 that direction rectification can serve as a plug-in component for existing magnitude-based methods. Finally, we compare the streaming efficiency of HIST3R with existing methods in Appendix A.4.

A.1 Local Structure in CUT3R State Evolution



(a) 7-Scenes (b) KITTI

(c) NRGBD (d) ScanNet

Figure 6: **Empirical local structure in CUT3R and HIST3R state evolution.** We visualize recurrent state evolution on one representative sequence from each of 7-Scenes [47], KITTI [50], NRGBD [48], and ScanNet [45]. In each subplot, the left panel shows the cosine affinity between recurrent states over time, while the right panel shows PCA analysis of state variations. CUT3R recurrent states naturally exhibit strong local affinity and low-dimensional variation, and HIST3R further accentuates this structure through history-guided rectification.

To further support the local-structure assumption used in HIST3R, we empirically analyze the recurrent state evolution of CUT3R during streaming inference. Given a sequence of recurrent states, we examine two complementary properties: the temporal affinity between states and the intrinsic dimensionality of their variations. Specifically, we compute the cosine affinity between recurrent states over time and apply PCA to the corresponding state differences. As shown in Fig. 6, representative sequences from 7-Scenes [47], KITTI [50], NRGBD [48], and ScanNet [45] consistently exhibit strong local structure. Adjacent CUT3R recurrent states have high cosine affinity, indicating that state evolution changes smoothly over short temporal windows rather than following arbitrary frame-wise transitions. Moreover, the PCA results show that state variations are largely concentrated in a small number of principal directions, suggesting that recent states trace a compact local state manifold. These observations provide empirical evidence for using recent history as a local reference for rectifying recurrent updates. In particular, when the CUT3R candidate endpoint is compatible with the local evolution pattern, it can be approximated by locally similar historical states, consistent with the self-expressiveness principle in low-dimensional subspace modeling [41]. HIST3R builds on this property by constructing a history-proposed endpoint from recent states and applying alignment-aware rectification, which further emphasizes the local structure already present in CUT3R state evolution.

Table 4: **Hyperparameter sensitivity.** Mean and standard deviation over all tested gate configurations of each component. For the last row, direction rectification uses the default setting while magnitude modulation varies. TUM-Dynamics ATE is evaluated at 1000 frames and KITTI scale-aligned AbsRel at 500 frames.

Method	TUM ATE ↓	KITTI AbsRel ↓
CUT3R	0.148	0.131
TTT3R	0.094	0.118
TTSA3R	0.084	0.117
Direction rect.	0.067 ± 0.008	0.118 ± 0.000
Magnitude mod.	0.093 ± 0.025	0.120 ± 0.001
Direction rect. (default) + Magnitude mod.	0.066 ± 0.009	0.117 ± 0.001

Table 5: **Direction rectification as a plug-in component.** We replace only the update direction of TTT3R and TTSA3R with the rectified direction of HIST3R while keeping their original magnitude controllers. ScanNet ATE is evaluated at 1000 frames and Bonn metric-depth AbsRel at 300 frames.

Method	ScanNet ATE ↓	Bonn AbsRel ↓
Direction rect. only	0.354	0.104
TTT3R	0.356	0.108
TTT3R + Direction rect.	**0.322**	**0.103**
TTSA3R	0.354	0.103
TTSA3R + Direction rect.	**0.337**	**0.097**

A.2 Hyperparameter Selection and Sensitivity

Selection protocol. All hyperparameters of HIST3R are selected once on 30 randomly sampled sequences from the ScanNet training split and kept fixed across all datasets, tasks, and sequence lengths. No test data or test performance is used for parameter selection. The final configuration uses a history window size of $k = 8$ and $(\lambda_{\text{align}}, \theta_{\text{align}}, \lambda_{\text{disc}}, \theta_{\text{disc}}) = (2, 0.94, 10, 0.24)$ throughout all experiments. The choice of k is analyzed in the design ablations in Sec. 4.4, where performance remains stable for $8 \leq k \leq 32$.

Sensitivity analysis. To verify that HIST3R does not rely on narrowly tuned gate parameters, we evaluate a broad range of configurations for each component. For direction rectification, we vary the gate sharpness $\lambda_{\text{align}} \in \{1, 2\}$ and threshold $\theta_{\text{align}} \in \{0.5, 0.6, 0.7, 0.8, 0.9\}$, resulting in 11 configurations including the default. For magnitude modulation, we vary $\lambda_{\text{disc}} \in \{1, 2, 10\}$ and $\theta_{\text{disc}} \in \{0.1, 0.2, 0.3, 0.4\}$, resulting in 13 configurations including the default. As shown in Table 4, direction rectification is stable across all tested configurations and consistently outperforms CUT3R, TTT3R, and TTSA3R on 1000-frame TUM-Dynamics. Magnitude modulation shows larger variation on this long-horizon setting, but adding the default direction rectification reduces both the mean error and its variation, from 0.093 ± 0.025 to 0.066 ± 0.009. These results indicate that the gains of HIST3R do not depend on specific gate parameters, and that direction rectification also stabilizes long-horizon updates.

A.3 Direction Rectification as a Plug-in Component

The direction–magnitude decomposition also allows our direction rectification to be combined with existing magnitude controllers. Under our decomposition, TTT3R and TTSA3R perform magnitude-only updates as in Eq. 3, where the gate $g_t^{(i)}$ is produced by each method and the candidate geometric intent $\tilde{d}_t^{(i)}$ is left unchanged. We replace only this direction with the rectified direction $\hat{d}_t^{(i)}$ of HIST3R while preserving each method's original gate:

$$S_t^{(i)} = S_{t-1}^{(i)} + g_t^{(i)} \tilde{m}_t^{(i)} \hat{d}_t^{(i)}. \tag{12}$$

As shown in Table 5, both hybrid variants outperform their original magnitude-only counterparts and direction rectification alone. This shows that direction rectification is independently effective rather than tied to a particular magnitude controller, and that it remains compatible with existing magnitude-based methods. The decomposition therefore not only unifies existing update-control methods, but also exposes geometric intent as a missing control axis that can be corrected independently and composed with different update-gain controllers.

A.4 Streaming Efficiency

We evaluate streaming efficiency on TUM-Dynamics [46] using a single NVIDIA H100 GPU with 80GB memory. We vary the number of input frames from 50 to 1000 and report the peak allocated GPU memory and inference speed, measured in frames per second (FPS).

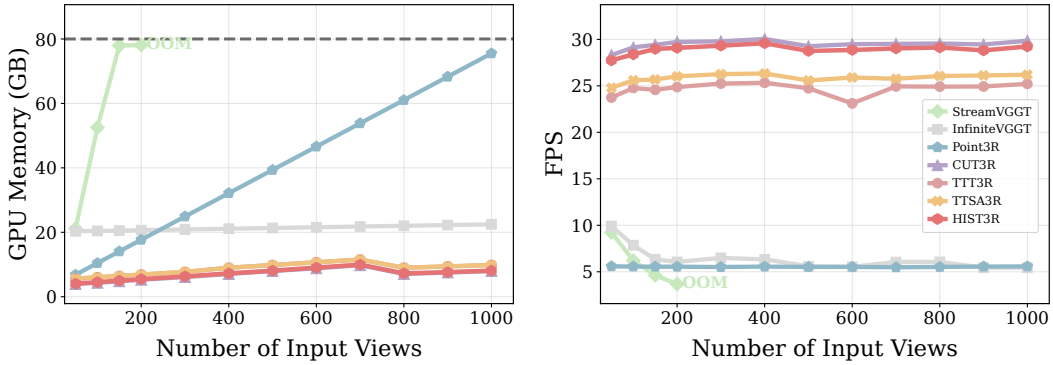


Figure 7: **Streaming efficiency on TUM-Dynamics [46].** Peak allocated GPU memory (left) and inference speed measured by FPS (right) are reported under increasing input sequence length.

Inference memory. As shown in Fig. 7, naive explicit-memory-based methods require increasingly large memory as the sequence length grows. StreamVGGT quickly exceeds the 80GB memory budget, while Point3R steadily accumulates memory due to its explicit pointmap memory and approaches the memory bound at 1000 frames. Although InfiniteVGGT mitigates this issue through cache pruning, it still requires substantially higher memory than implicit-memory-based methods. In contrast, CUT3R-style implicit-memory methods, including HIST3R, maintain the lowest and nearly constant memory usage across different sequence lengths, since historical information is compressed into a compact recurrent state rather than stored as an ever-growing explicit memory.

Inference speed. In terms of inference speed, explicit-memory-based methods are substantially slower because they rely on attention over large explicit memories or caches. By maintaining historical information in a compact recurrent state, implicit-memory-based methods use much smaller attention matrices and achieve faster streaming inference. Among CUT3R-based extensions, HIST3R maintains nearly identical FPS to the original CUT3R and is substantially faster than TTT3R and TTSA3R. HIST3R only exposes the state-to-image attention map from the first decoder layer for constructing visual summaries, allowing the remaining decoder layers to retain FlashAttention [51]. By contrast, TTT3R and TTSA3R require attention maps from all decoder layers, which materializes attention matrices throughout inference and introduces significant overhead. These results show that HIST3R improves long-sequence reconstruction quality while preserving the memory and speed advantages of compact recurrent state-space inference.

B More Results on Additional Settings

The main paper primarily focuses on long-sequence evaluation, where streaming methods are tested on their ability to maintain a stable scene representation over hundreds of frames. Here, we provide additional evaluations on sparse-view reconstruction and standard short-sequence benchmarks to further assess the generalization ability of HIST3R across different input regimes. The sparse-view setting tests robustness under widely separated observations with limited cross-view overlap, while

the short-sequence setting enables comparison with a wider range of baselines that are infeasible for long-sequence streaming due to memory constraints. Together with the long-sequence results in the main paper, these evaluations show that the gains of HIST3R are concentrated in long-horizon streaming, while it remains competitive on standard short sequences and preserves most of the sparse-view robustness of CUT3R.

B.1 Sparse-View Reconstruction

Table 6: **Sparse-view 3D reconstruction results on 7-Scenes [47] and NRGBD [48]**. Following CUT3R [23], we sparsely sample images with stride 200 for 7-Scenes and stride 500 for NRGBD. Best results are shown in bold, and second-best results are underlined.

Method	Input	7-Scenes (Stride 200)						NRGBD (Stride 500)					
		Acc. ↓		Comp. ↓		NC ↑		Acc. ↓		Comp. ↓		NC ↑	
		Mean	**Med.**	**Mean**	**Med.**	**Mean**	**Med.**	**Mean**	**Med.**	**Mean**	**Med.**	**Mean**	**Med.**
CUT3R	*Sparse View*	**0.120**	**0.047**	**0.156**	**0.030**	**0.725**	0.831	**0.100**	0.036	**0.078**	0.031	0.834	**0.970**
TTT3R		0.156	0.054	0.195	0.046	0.719	0.824	0.105	0.032	0.080	0.027	0.835	**0.970**
TTSA3R		0.152	0.054	0.189	0.039	0.718	0.821	0.107	0.033	0.082	0.028	0.834	0.968
HIST3R		0.147	0.049	0.176	0.031	**0.725**	**0.834**	0.104	**0.031**	0.080	**0.026**	**0.836**	**0.970**

Although HIST3R is designed for long-sequence streaming reconstruction, we further evaluate its robustness under sparse-view reconstruction. Following CUT3R [23], we evaluate scene-level reconstruction using sparsely sampled images: 3 to 5 frames per scene for 7-Scenes [47] and 2 to 4 frames per scene for NRGBD [48]. We assess reconstruction quality by comparing predicted pointmaps with ground-truth point clouds using accuracy, completeness, and normal consistency (NC). Accuracy measures the nearest-neighbor distance from the predicted reconstruction to the ground truth, while completeness measures the distance in the reverse direction. For all three metrics, we report both mean and median values.

This setting is particularly challenging for history-guided updates, since sparse inputs induce large viewpoint changes with limited overlap and weaken the local temporal structure that HIST3R relies on. As shown in Table 6, vanilla CUT3R achieves the best overall performance, and all training-free extensions introduce some degradation. Among them, HIST3R consistently outperforms TTT3R and TTSA3R and remains closest to CUT3R, matching or outperforming it on 6 of the 12 metrics, including normal consistency on both datasets. These results indicate that HIST3R is most beneficial in dense streaming settings, while it largely preserves the robustness of CUT3R when historical structure is less reliable. They also show that its long-horizon gains do not come from merely conservative state updates.

B.2 Standard Short-Sequence Evaluation

Baselines. We include all baselines evaluated in the main paper, covering CUT3R, its training-free extensions, and explicit-memory-based methods such as Point3R, StreamVGGT, and InfiniteVGGT. To broaden the short-sequence comparison, we further include offline 3D foundation models. Pairwise methods, including DUSt3R [11], MASt3R [12], MonST3R [28], and Easi3R [32], take image pairs as input and require a global alignment stage to consolidate pairwise predictions. We also compare with AETHER [52] and VGGT [13], which predict all pointmaps jointly and achieve strong short-sequence reconstruction performance. These offline methods are effective for short sequences but are not designed for streaming inference, since adding a new frame requires rerunning inference or global alignment over the sequence. For online baselines beyond those in the main paper, we additionally include STream3R [21] and Spann3R [38]. STream3R fine-tunes DUSt3R with a causal KV-cache for streaming reconstruction, while Spann3R extends DUSt3R to online reconstruction with an external spatial memory that stores key-value features of previous frames.

Camera pose estimation. Following prior works [23, 24, 28], we evaluate camera pose estimation on Sintel [53], TUM-Dynamics [46], and ScanNet [45]. We report Absolute Trajectory Error (ATE), Relative Translation Error (RTE), and Relative Rotation Error (RRE), after aligning the estimated camera trajectory to the ground truth with Sim(3) alignment following the standard evaluation protocol [46]. In addition to the streaming baselines considered in the main paper, we include a

Table 7: **Evaluation on Camera Pose Estimation - Short Sequence** on Sintel [53], TUM-dynamics [46], and ScanNet [45] datasets. HIST3R achieves the best overall performance among online methods and remains competitive with most offline baselines, despite processing frames incrementally rather than reasoning over the full sequence at once.

		Sintel (50 frames)			TUM-dynamics (90 frames)			ScanNet (90 frames)		
Method	**Online**	ATE ↓	RPE trans ↓	RPE rot ↓	ATE ↓	RPE trans ↓	RPE rot ↓	ATE ↓	RPE trans ↓	RPE rot ↓
Robust-CVD [54]	✗	0.360	0.154	3.443	0.153	0.026	3.528	0.227	0.064	7.374
CasualSAM [55]	✗	0.141	**0.035**	0.615	0.071	**0.010**	1.712	0.158	0.034	1.618
DUSt3R [11]	✗	0.417	0.250	5.796	0.083	0.017	3.567	0.081	0.028	0.784
MASt3R [12]	✗	0.185	0.060	1.496	0.038	0.012	0.448	0.078	0.020	0.475
MonST3R [28]	✗	0.111	0.044	0.869	0.098	0.019	0.935	0.077	0.018	0.529
Easi3R [32]	✗	**0.110**	0.042	0.758	0.105	0.022	1.064	0.061	0.017	0.525
AETHER[52]	✗	0.189	0.054	0.694	0.092	0.012	1.106	0.176	0.028	1.204
VGGT [13]	✗	0.172	0.062	**0.471**	**0.012**	**0.010**	**0.310**	**0.035**	**0.015**	**0.377**
Point3R [19]	✓	0.351	0.128	1.822	0.075	0.029	0.642	0.106	0.035	1.946
STream3R [21]	✓	0.213	0.076	0.868	0.026	0.013	0.330	**0.052**	0.021	0.850
StreamVGGT [20]	✓	0.251	0.149	1.894	0.061	0.033	3.209	0.161	0.057	3.647
InfiniteVGGT [22]	✓	0.227	0.130	1.669	0.054	0.031	2.796	0.146	0.052	3.290
Spann3R [38]	✓	0.329	0.110	4.471	0.056	0.021	0.591	0.096	0.023	0.661
CUT3R [23]	✓	0.213	0.066	0.621	0.046	0.015	0.473	0.099	0.022	0.600
TTT3R [24]	✓	0.201	**0.063**	**0.617**	0.028	0.012	0.379	0.064	0.021	0.592
TTSA3R [25]	✓	0.210	0.085	0.765	0.028	0.012	0.372	0.057	0.020	0.588
HIST3R	✓	**0.195**	0.080	0.748	**0.023**	**0.011**	**0.329**	0.057	**0.019**	**0.525**

broader set of short-sequence baselines, including 3D reconstruction foundation models and per-sequence optimization methods such as Robust-CVD [54] and CasualSAM [55], which jointly optimize camera parameters and dense depth maps for each sequence.

The short-sequence results are reported in Table 7. Since several offline baselines rely on global optimization or full-sequence attention, they are effective in this short-sequence setting but are not directly applicable to long-sequence streaming evaluation due to memory constraints. We therefore report offline and online methods separately. Offline methods can access all input images simultaneously and perform global reasoning over the entire sequence, whereas online methods must process images incrementally, creating an inherent gap between the two settings. Although HIST3R does not match the performance of the state-of-the-art offline method VGGT, it achieves the strongest overall performance among online methods. Notably, HIST3R also surpasses several offline baselines, showing that its design remains highly competitive even under standard short-sequence evaluation.

Video Depth Estimation Following common practice [23, 24, 28], we evaluate video depth estimation on KITTI [50], Sintel [53], and Bonn [49] datasets covering dynamic and static, indoor and outdoor scenes. We report absolute relative error (Abs Rel) and $\delta < 1.25$, which measures the percentage of pixels whose predicted depth is within a factor of 1.25 of the ground truth. To evaluate relative depth quality and temporal consistency, we align predicted depths to the ground truth with a single scale factor per sequence. For methods that output metric pointmaps, we additionally report results without scale alignment, directly evaluating the predictions in metric units to measure absolute-scale accuracy.

As shown in Table 8, HIST3R achieves the strongest overall metric-scale performance. Under scale-aligned evaluation, HIST3R remains competitive with leading online methods such as StreamVGGT and InfiniteVGGT on Bonn and KITTI, while being substantially more efficient in both memory usage and inference speed. Together with the long-sequence and sparse-view results, these results show that the benefits of HIST3R are not limited to long-horizon streaming reconstruction; its update design also remains effective under standard short-sequence evaluation and challenging sparse-input scenarios.

C Qualitative Results

We provide qualitative results for the long-horizon streaming reconstruction setting studied in the main paper. Specifically, we visualize both 3D reconstruction and camera pose estimation results under

Table 8: **Evaluation of video depth estimation on short sequences** on Sintel [53], Bonn [49], and KITTI [50]. We report scale-invariant relative depth, aligned by a per-sequence scale, and metric-scale absolute depth accuracy. HIST3R achieves the strongest metric-scale performance, while remaining competitive with leading online methods under scale-aligned evaluation on Bonn and KITTI.

			Sintel (50 frames)		**BONN (110 frames)**		**KITTI (110 frames)**	
Alignment	**Method**	**Online**	Abs Rel ↓	δ<1.25 ↑	Abs Rel ↓	δ<1.25 ↑	Abs Rel ↓	δ <1.25 ↑
Per-sequence Scale	DUSt3R [11]	✗	0.656	45.2	0.155	83.3	0.144	81.3
	MASt3R [12]	✗	0.641	43.9	0.252	70.1	0.183	74.5
	MonST3R [28]	✗	0.378	55.8	0.067	96.3	0.168	74.4
	Easi3R [32]	✗	0.377	55.9	0.059	97.0	0.102	91.2
	AETHER [52]	✗	0.324	50.2	0.273	59.4	**0.056**	**97.8**
	VGGT [13]	✗	**0.287**	**66.1**	**0.055**	**97.1**	0.070	96.5
	Point3R [19]	✓	0.452	48.9	0.060	96.0	0.136	84.2
	STream3R [21]	✓	0.478	51.1	0.075	94.1	0.116	89.6
	StreamVGGT [20]	✓	**0.323**	**65.7**	**0.059**	97.2	0.173	72.1
	InfiniteVGGT [22]	✓	0.327	**65.7**	0.060	**97.3**	0.173	72.1
	Spann3R [38]	✓	0.622	42.6	0.144	81.3	0.198	73.7
	CUT3R [23]	✓	0.421	47.9	0.078	93.7	0.118	88.1
	TTT3R [24]	✓	0.404	50.0	0.068	95.4	0.113	90.4
	TTSA3R [25]	✓	0.401	50.0	0.064	96.5	**0.110**	**91.2**
	HIST3R	✓	0.408	50.1	0.061	96.9	**0.110**	90.8
Metric Scale	MASt3R [12]	✗	1.022	14.3	0.272	70.6	0.467	15.2
	Point3R [19]	✓	**0.777**	17.1	0.137	94.7	0.191	73.8
	STream3R [21]	✓	1.041	21.0	0.084	94.4	0.234	57.6
	CUT3R [23]	✓	1.029	23.8	0.103	88.5	0.122	85.5
	TTT3R [24]	✓	0.977	24.5	0.090	94.2	**0.110**	89.1
	TTSA3R [25]	✓	0.960	24.6	0.079	96.6	0.111	88.9
	HIST3R	✓	0.935	**25.3**	**0.068**	**96.7**	**0.110**	**90.8**

500-frame inputs, which represent a challenging sequence length for existing online reconstruction methods.

3D Reconstruction. We provide qualitative comparisons on 7-Scenes and NRGBD under the 500-frame streaming setting. To make the comparison clearer, we organize the results according to the type of online memory used by the baselines: implicit-memory methods and explicit-memory methods. For implicit-memory methods, Fig. 8 and Fig. 9 show that CUT3R suffers from severe long-horizon degradation, leading to blurred and inconsistent reconstructions. TTT3R partially alleviates this degradation, but still exhibits visible artifacts and structural drift. In contrast, HIST3R produces cleaner and more stable reconstructions, demonstrating the benefit of direction–magnitude update control for maintaining scene geometry over long sequences. We further compare with explicit-memory-based methods in Fig. 10 and Fig. 11. Point3R can produce distorted scene geometry in some cases. HIST3R remains competitive with InfiniteVGGT while preserving stable scene structure, suggesting its ability to maintain strong long-horizon reconstruction quality without relying on explicit memory storage.

Camera Pose Estimation. We provide qualitative trajectory comparisons on ScanNet and TUM-Dynamics in Fig. 12 and Fig. 13. Existing online baselines often accumulate visible trajectory drift over long sequences. HIST3R, in contrast, closely follows the ground-truth trajectory, demonstrating improved long-horizon pose stability under streaming inference.

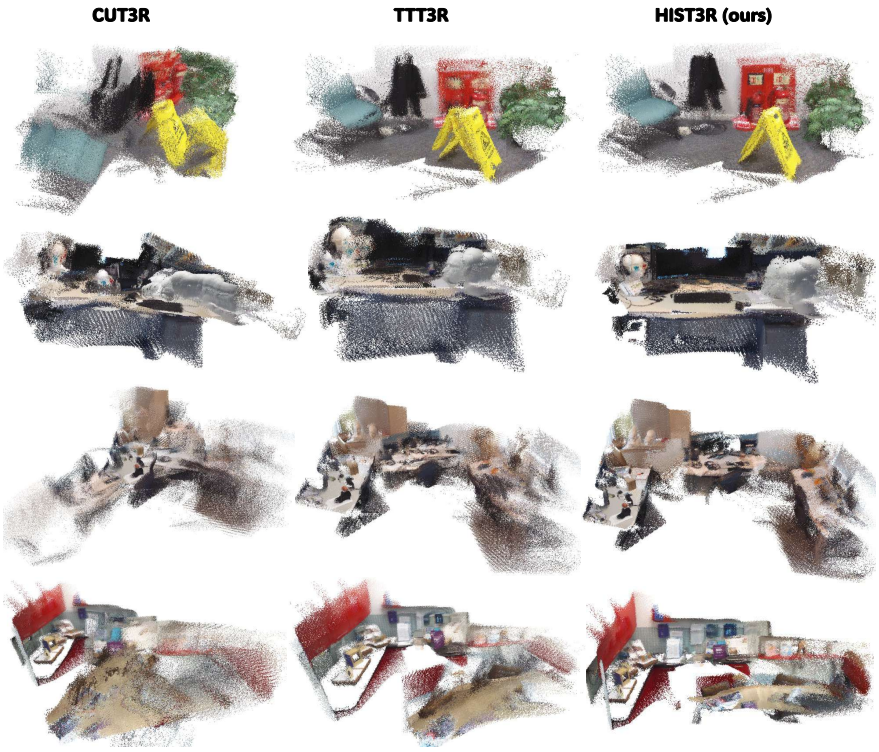


Figure 8: **Qualitative comparison with implicit-memory-based methods on 7-Scenes.** CUT3R exhibits clear long-horizon degradation, while HIST3R substantially improves reconstruction coherence and reduces accumulated artifacts.

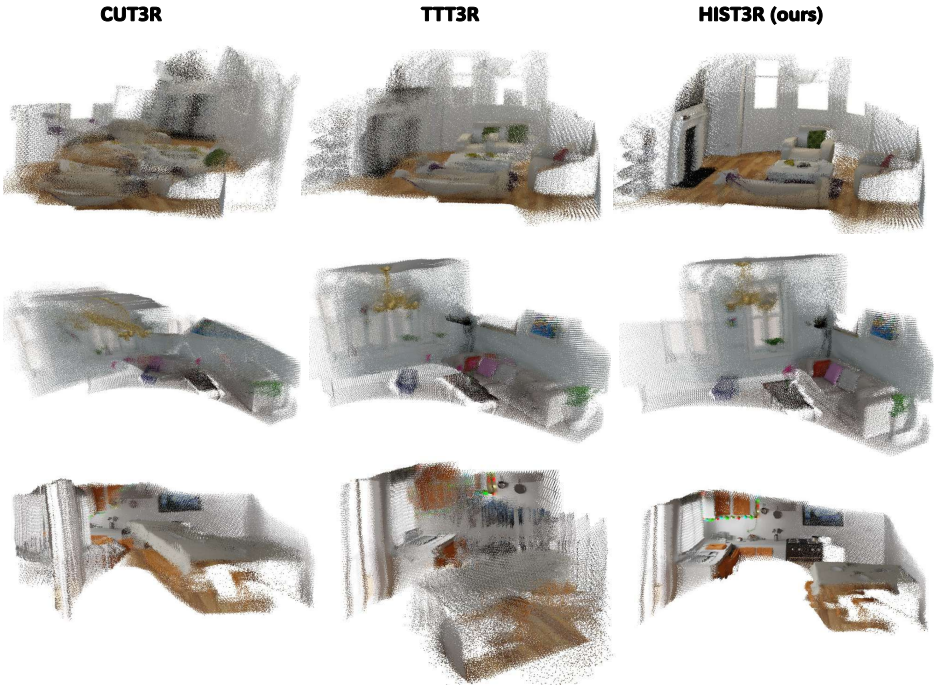


Figure 9: **Qualitative comparison with implicit-memory-based methods on NRGBD.** TTT3R partially improves over CUT3R, but still exhibits severe accumulated artifacts and collapsed scene geometry, whereas HIST3R produces substantially more coherent and stable reconstructions.

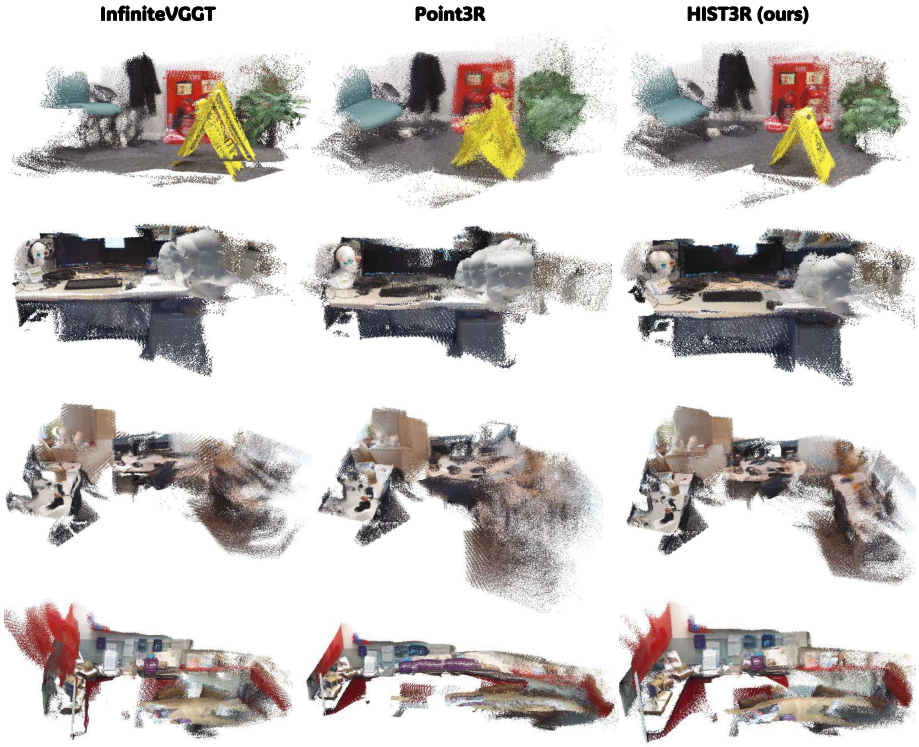


Figure 10: **Qualitative comparison with explicit-memory-based methods on 7-Scenes.** Point3R exhibits noticeable geometric distortion, while InfiniteVGGT remains competitive with HIST3R but still produces slightly blurred reconstructions in some regions.

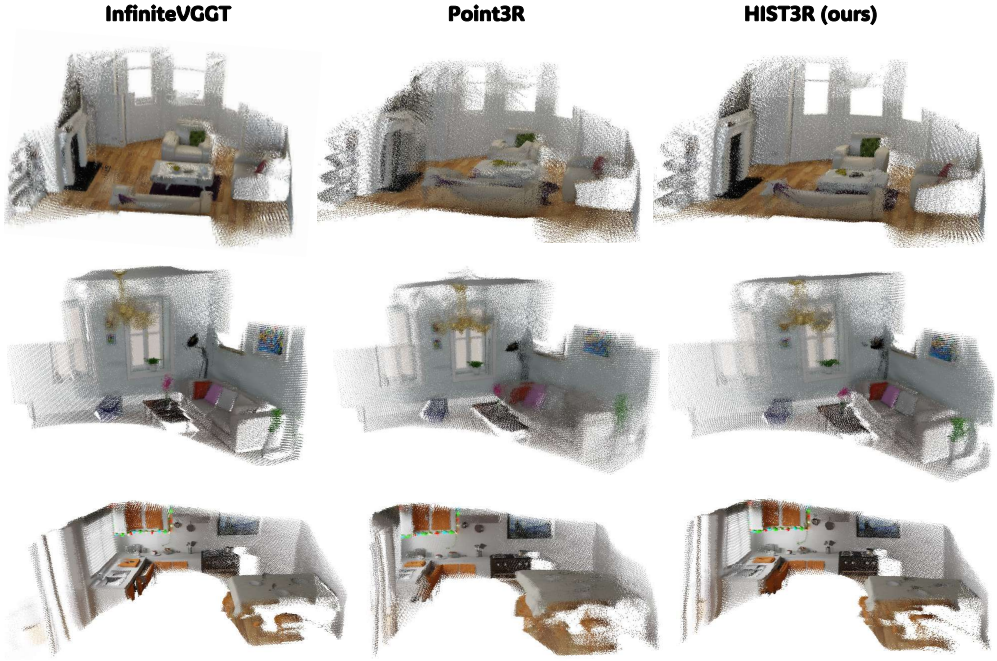


Figure 11: **Qualitative comparison with explicit-memory-based methods on NRGBD.** HIST3R and InfiniteVGGT produce more stable reconstruction quality than Point3R, with cleaner local geometry and fewer visible distortions.

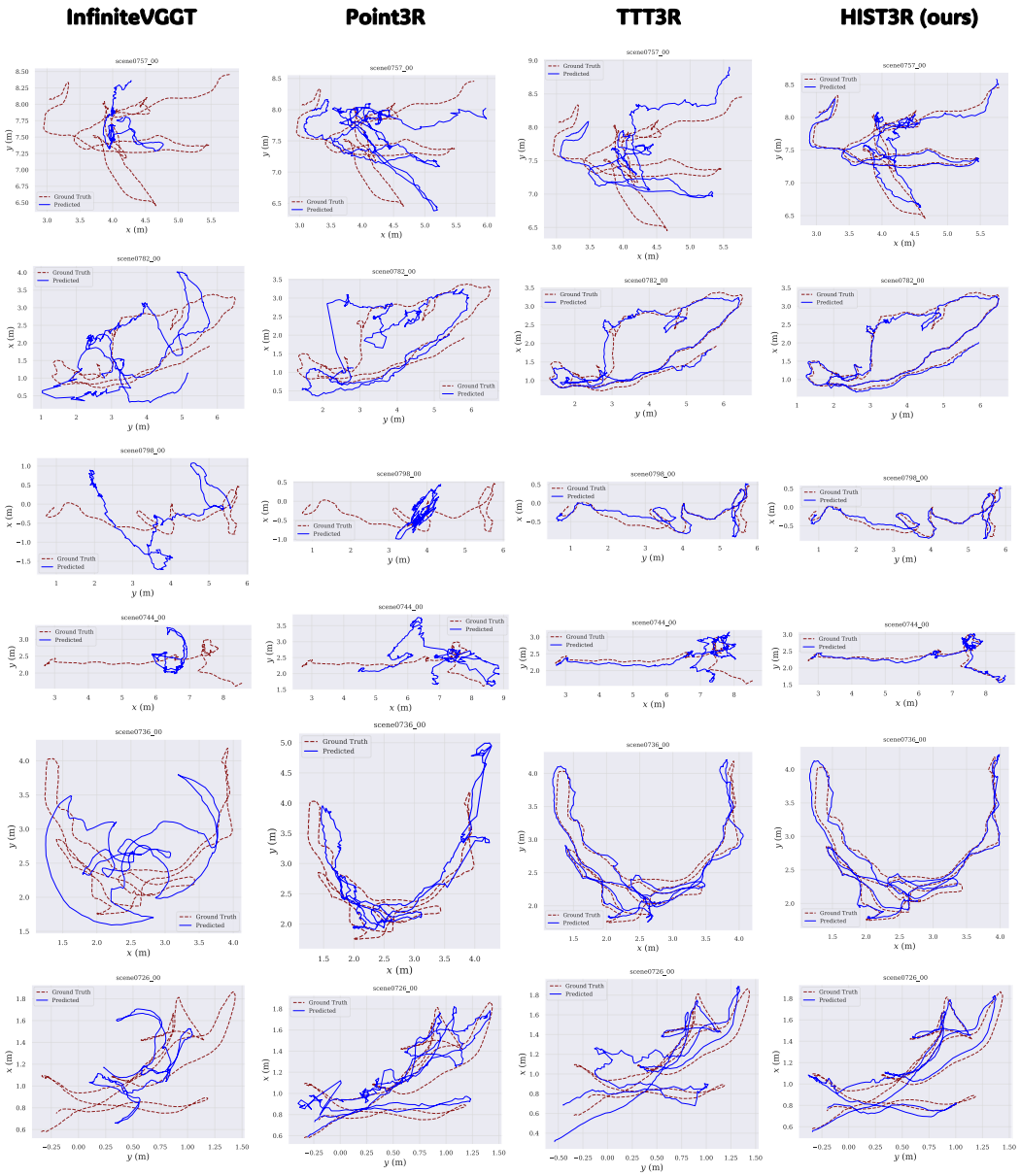


Figure 12: **Qualitative trajectory results on ScanNet.** Blue curves denote predicted trajectories, and red dashed curves denote ground-truth trajectories. Explicit-memory-based methods exhibit noticeable pose drift over long sequences. TTT3R produces more stable trajectories, but still shows a clear gap compared with HIST3R, which better preserves long-horizon trajectory consistency.

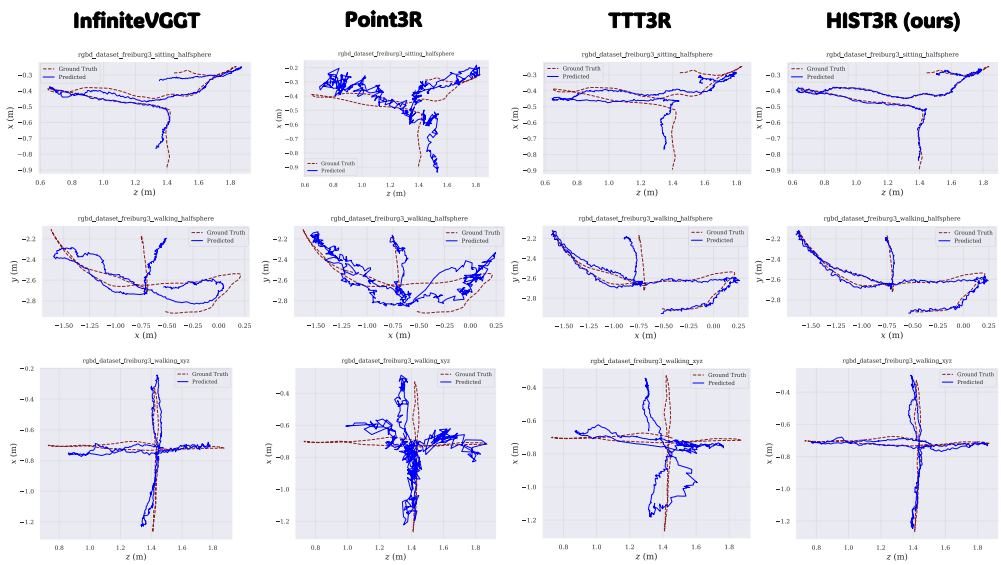


Figure 13: **Qualitative trajectory results on TUM-Dynamics.** HIST3R shows a clear advantage over all compared methods, preserving substantially better long-horizon trajectory consistency.

D Discussion

Limitations. As a training-free method, HIST3R is ultimately bounded by the representation and candidate updates produced by the pretrained CUT3R model. Although HIST3R improves recurrent state updates by modulating the direction and magnitude of each candidate update at inference time, it cannot recover geometric information that is not encoded by the underlying model or fully correct failures caused by severely inaccurate candidate predictions. A promising direction is to incorporate this direction–magnitude view into pretraining or fine-tuning, so that the model can learn to produce more reliable geometric intent and update gain rather than correcting them only at inference time.

Broader impact. This work analyzes the evolution of implicit recurrent states in streaming 3D reconstruction from a geometric perspective. By decomposing state updates into direction and magnitude, HIST3R provides an interpretable view of how online reconstruction models accumulate, preserve, and revise geometric information over long sequences. We believe this perspective can inform the design of future state-space reconstruction models, encouraging update mechanisms that explicitly reason about both geometric intent and update strength. Such models may benefit online 3D perception systems that require stable long-horizon reconstruction under bounded memory, including embodied AI, augmented reality, and robotics. At the same time, advances in streaming reconstruction may strengthen downstream applications involving spatial understanding and scene capture, and should therefore be deployed with appropriate consideration of privacy and safety when reconstructing real-world environments.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes].

   Justification: The abstract and introduction accurately describe HIST3R as a training-free extension of CUT3R that decomposes recurrent updates into direction and magnitude. The stated claims are supported by experiments across multiple 3D perception tasks, and the scope is clearly limited to inference-time improvement over pretrained streaming reconstruction models.

   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: We discuss limitations in Appendix D, noting that HIST3R is a training-free method whose performance is bounded by the pretrained CUT3R model, and that future work could incorporate the direction–magnitude decomposition into pretraining or fine-tuning. We also discuss in Appendix B.1 that HIST3R is most beneficial in dense streaming settings and slightly underperforms CUT3R under sparse-view inputs, where the local temporal structure it relies on is weaker.

   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 present formal theoretical results or proofs. The self-expressive and direction–magnitude decomposition views are used as methodological motivation and are supported empirically through ablations and analysis.

   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: We describe the evaluation protocols, datasets, baselines, and metrics in Sec. 4.1–4.4, and the full method in Sec. 3. The selection protocol and final values of all hyperparameters, including the history window size and the alignment and discrepancy gate parameters, are reported in Appendix A.2, together with a sensitivity analysis. Design ablations are provided in Sec. 4.4.

   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: [No].

   Justification: The code is not publicly released at the time of publication. All experiments use publicly available datasets, and we describe the evaluation protocols, metrics, baselines, and hyperparameters needed to reproduce the main results.

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes].

   Justification: The paper specifies the test protocols, datasets, metrics, and baselines in Sec. 4.1–4.4. All hyperparameters are selected once on 30 randomly sampled sequences from the ScanNet training split without using test data, and kept fixed across all experiments; their values and selection protocol are reported in Appendix A.2. Since HIST3R is training-free, no optimizer or training schedule is required.

   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: We do not report error bars or statistical significance tests. Our evaluation follows prior streaming reconstruction work and reports deterministic benchmark metrics across multiple datasets, sequence lengths, and tasks.

Guidelines:

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

8. **Experiments compute resources**

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

   Answer: [Yes].

   Justification: The GPU type and memory budget are reported in Sec. 4 and Appendix A.4, where all experiments are conducted on a single NVIDIA H100 GPU with 80GB memory. Since HIST3R is training-free, no training compute is required; inference-time memory and speed are reported in Appendix A.4 and Fig. 7.

   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 research uses standard public 3D reconstruction benchmarks and does not involve human-subject studies, private data collection, or deployment on sensitive real-world systems. We also discuss broader impact and potential privacy considerations for real-world reconstruction applications.

Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes].

    Justification: The paper discusses broader impacts in the appendix, including potential benefits for online 3D perception systems such as robotics, embodied AI, and AR/VR, as well as privacy and safety considerations when reconstructing real-world environments.

    Guidelines:

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

11. **Safeguards**

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

    Answer: [N/A].

    Justification: The paper does not release new high-risk datasets or pretrained generative models. The method is evaluated on standard public 3D reconstruction benchmarks and builds on existing reconstruction models.

    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: The paper cites the original works for all datasets, baselines, and pretrained models used in the experiments. We use standard public benchmarks and existing models according to their intended research use.

    Guidelines:

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

13. **New assets**

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

    Answer: [N/A].

    Justification: The paper does not introduce new datasets or benchmark assets. HIST3R is a training-free method evaluated on existing public datasets and baselines.

    Guidelines:

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

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

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

    Answer: [N/A].

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

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

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

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

    Answer: [N/A].

    Justification: The paper does not involve research with human subjects, crowdsourcing, or participant studies; therefore, 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 are not used as an important, original, or non-standard component of the core method. Any LLM usage, if applicable, is limited to writing, editing, or formatting assistance and does not affect the methodology, experiments, or scientific claims.

    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.