FAST AND STABLE LONG-CONTEXT 3D RECONSTRUCTION USING RECON-AWARE MEMORY BANK

Anonymous authors
Paper under double-blind review

ABSTRACT

Feed-forward 3D reconstruction models such as VGGT provide high-quality geometry and camera estimation without per-scene optimization, but scaling them to long visual sequences remains challenging. Dense models incur quadratic attention computation as context grows, while streaming variants still accumulate increasingly large historical context. We argue that efficient context reduction should be both **reconstruction-aware** and **architecture-decoupled**: selected frames should preserve downstream reconstruction quality without relying on backbone-specific cache or attention structures. To this end, we introduce **Reconstruction-Aware Memory Bank (RAMB)**, a plug-in framework that uses frame-associated scene tokens as a common representation across VGGT-style backbones. RAMB combines a lightweight Rank Model trained with downstream reconstruction outcomes to identify high-utility frames with scene-token similarity to retain complementary observations and remove redundant context.

We evaluate RAMB on VGGT-Ω and StreamVGGT across ScanNet, ETH3D, and KITTI-VO. The selected frames are highly informative: reconstructing from only about 3% of a 400-frame sequence, VGGT-Ω matches or exceeds full-sequence reconstruction accuracy (ACC). For chunked inference, RAMB carries informative context across chunks, enabling 540-frame reconstruction on a single 24 GB GPU, where full-sequence inference requires over 60 GB, with comparable quality. For StreamVGGT, RAMB bounds the historical context to 20.9 of 240 frames on average while maintaining comparable quality, and reduces forward latency and peak memory by $4\times$.

1 INTRODUCTION

Feed-forward 3D reconstruction has emerged as a powerful alternative to traditional optimization-based pipelines (Wang & Agapito, 2025; Wang et al., 2025b). Methods such as VGGT (Wang et al., 2025a) and VGGT-Ω (Wang et al., 2026) directly infer 3D geometry, camera poses, and depth from unposed images without costly per-scene optimization (Schonberger & Frahm, 2016; Wu et al., 2011; Kerbl et al., 2023; Mildenhall et al., 2021), making them attractive visual-geometry backbones for robotics, autonomous systems, and spatial computing.

A major challenge is scaling these models to long visual sequences. Applications such as egocentric capture and AR/VR naturally produce hundreds or thousands of frames (Pan et al., 2023; Grauman et al., 2022), while the computation and memory cost of feed-forward reconstruction grows rapidly with sequence length. Dense-input models such as VGGT and VGGT-Ω jointly process all views, whereas streaming variants such as StreamVGGT avoid recomputing the entire history but still accumulate historical representations. As a result, even recent models remain constrained to relatively short sequences under consumer-level GPU memory budgets (e.g. 24GB VRAM).

Prior work has explored selective context processing, e.g., IncVGGT (Fang et al., 2026) consolidates input frames and Evict3R (Mahdi et al., 2025) removes redundant cache entries. However, two key challenges remain unresolved for effective long-context reconstruction:

- **Reconstruction-aware frame selection:** Frame utility should reflect its contribution to the final reconstruction, yet existing methods often rely on proxy signals such as overlap, recency, attention scores, or KV-cache statistics.

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

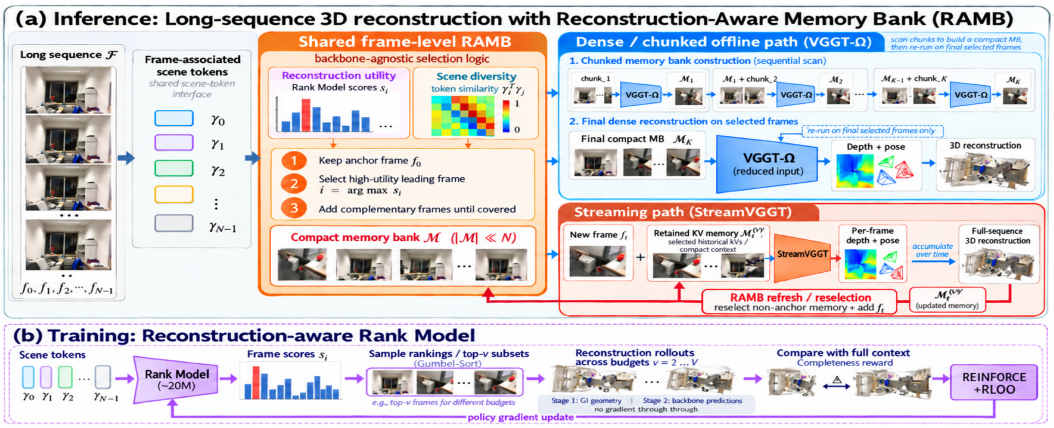

Figure 1: **Overview of the RAMB pipeline**

- **Architecture-decoupled context management:** Many context-management methods depend on backbone-specific attention patterns, internal states, or KV-cache organization, limiting portability as architectures evolve.

We address these challenges with **Reconstruction-Aware Memory Bank (RAMB)**, a plug-in context-management framework for long-sequence 3D reconstruction. Our key observation is that VGGT-family models expose compact, frame-associated **scene tokens** that provide a natural interface for context selection. RAMB uses these tokens in two complementary ways: a lightweight **Rank Model**, trained with reconstruction rollouts, estimates frame utility from downstream reconstruction quality, while scene-token similarity identifies complementary views and suppresses redundant context.

Because RAMB operates on frame-level scene representations rather than model-specific KV-cache states, the same framework can be applied to different VGGT-style backbones with only minor adaptation. For dense-input VGGT-Ω, RAMB selects a compact subset of frames as the effective input for reconstruction; for streaming-input StreamVGGT, it maintains a compact historical context while continuing to process all incoming observations. More precisely, RAMB can be applied to a backbone as long as the following two prerequisites are satisfied: (1) The backbone contains attention-based hidden layers with register tokens. (2) The register tokens maintain a correspondence with the input frames, and their representations capture inter-frame content similarity.

We evaluate RAMB on long sequences from ScanNet, ETH3D, and KITTI-VO. Across VGGT-Ω and StreamVGGT, RAMB reduces retained context by approximately $30\times$ and $10\times$, respectively, while maintaining reconstruction quality comparable to full-context processing. This enables substantially longer sequences under the same GPU memory budget and reduces forward inference latency by $2.25\times$ and $4\times$, respectively. Our results further show that RAMB benefits not only from shorter effective context, but also from selecting more informative frames than uniform sampling.

In summary, we make the following contributions:

- We formulate long-context feed-forward 3D reconstruction as a **reconstruction-aware context-selection problem**, where the goal is to retain a compact subset of frames that preserves the reconstruction supported by the full context.
- We introduce a **rollout-trained frame-selection mechanism** that learns reconstruction utility from downstream quality and combines it with scene-token similarity to balance utility and scene coverage.
- We develop **RAMB**, an architecture-decoupled frame-level memory-bank framework that applies to both dense and streaming VGGT-style models and reduces retained context by more than $10\times$ while preserving reconstruction quality.

2 Related Work

VGGT Model Family: VGGT is a feed-forward 3D reconstruction model that directly predicts geometry, appearance, camera poses, and per-pixel depth in a single forward pass. Follow-up work includes StreamVGGT, which extends VGGT to streaming inputs, and its next-generation successor, VGGT-Ω. We focus on VGGT-Ω and StreamVGGT as representative state-of-the-art models

for dense-input and streaming-input reconstruction, respectively. Although both improve the memory, latency, and long-context limitations of VGGT, substantial scalability challenges remain. On a 24 GB consumer GPU, StreamVGGT supports up to 80 frames at below 2 FPS near this limit, while VGGT-Ω processes up to 220 frames with approximately 35 s latency. Moreover, reconstruction quality still degrades with increasing sequence length (Section 4.3.2), leaving long-context reconstruction an open challenge.

Memory Bank: Memory banks are widely used for long-term dependency modeling in video understanding, tracking, and 3D reconstruction (Ravi et al., 2025; Zhou et al., 2024). Recent VGGT-based methods primarily manage internal KV caches: XStreamVGGT (Su et al., 2026) compresses them via dimension-adaptive quantization, InfiniteVGGT (Yuan et al., 2026) prunes tokens to a fixed budget, RetrieveVGGT (Zou et al., 2026) retrieves history by query-key similarity, and Mem3R (Liu et al., 2026) combines hybrid memory with test-time training. These approaches are tightly coupled to model-specific attention patterns and KV-cache organization, including assumptions about K/V distributions, pruning granularity, and descriptors. As foundation architectures evolve, such mechanisms therefore would require substantial redesign to accommodate new backbones.

Learning Frame Redundancy from Rollouts: Parallel work in LLMs addresses growing KV caches through learned eviction. Moschella et al. (2026) formulate eviction as reinforcement learning, using downstream rollouts to measure the realized utility of retaining or discarding cached tokens. We adopt the same principle for 3D reconstruction: reconstruction rollouts over candidate frame subsets reveal frames whose geometric evidence is redundant. Removing such frames preserves reconstruction quality while shortening the input, thereby reducing latency and memory use and mitigating backbone degradation on long sequences.

3 METHOD

We provide an overview of VGGT and StreamVGGT in Section 3.1, then describe our proposed Rank Model in Section 3.2 and the frame selection method in Section 3.3. Finally, we summarize these approaches and unify them under a general Memory Bank framework in Section 3.4.

3.1 PRELIMINARIES

VGGT-Style Feed-forward 3D Reconstruction: Given a set of N RGB frames $\mathcal{F} = \{f_i\}_{i=0}^{N-1}$, $f_i \in \mathbb{R}^{H\times W\times 3}$, VGGT-family models directly infer multi-view geometry in a feed-forward manner. Each image is encoded into patch tokens P_i, augmented with learned camera and register tokens, and processed by alternating frame-wise and cross-frame attention blocks. We abstract the backbone model[1] as $BM(\mathcal{F}) \to \{(\hat{\mathbf{c}}_i, \hat{D}_i)\}_{i=0}^{N-1}$, where $\hat{\mathbf{c}}_i$ and $\hat{D}_i$ denote the estimated camera parameters and dense depth of frame i. Following the VGGT convention, the anchor frame f_0 serves as the reference view, and 3D points are obtained by back-projecting the predicted depth with the estimated cameras.

The two backbones differ in how they execute cross-frame computation. Dense models such as VGGT-Ω process all frames jointly, $\{(\hat{\mathbf{c}}_i, \hat{D}_i)\}_{i=0}^{N-1} = BM_{\text{dense}}(\mathcal{F})$. Incorporating a new frame, however, requires another joint forward pass over the enlarged input. StreamVGGT instead makes cross-frame attention causal and reuses historical KVs. At layer ℓ, the current frame is processed as $X_t^{\ell+1} = \text{Attn}\left(Q_t^\ell, [K_{<t}^\ell; K_t^\ell], [V_{<t}^\ell; V_t^\ell]\right)$, where $(K_{<t}^\ell, V_{<t}^\ell)$ are cached from previous frames. The cache is updated as $\mathcal{M}_t^\ell = \mathcal{M}_{t-1}^\ell \cup \{K_t^\ell, V_t^\ell\}$. This enables incremental inference without recomputing previous frames, but memory and attention costs keep growing with the retained history.

Register and Scene Tokens: For each frame, the backbone representation can be written as $X_i = [C_i;\, R_i;\, P_i]$, where P_i are patch tokens, C_i is the camera token, and R_i denotes a small set of register tokens. StreamVGGT keeps these registers but, under temporal causal attention, caches them as a token memory that carries scene information from past frames forward in time. VGGT-Ω further expands these registers and explicitly treats them as *scene tokens*, which serve as compact carriers of scene-level information. Besides full global attention over all frame tokens, VGGT-Ω introduces register attention, $\{R_i'\}_{i=0}^{N-1} = \text{Attn}_{\text{reg}}\left(\{R_i\}_{i=0}^{N-1}\right)$, which exchanges information across frames

[1] **Backbone Model:** As this work primarily focuses on VGGT-Ω and StreamVGGT, we collectively refer to these two models as the "backbone model" (BM) throughout the paper, unless otherwise specified.

through the compact register tokens. Subsequent frame attention propagates the aggregated scene information back to the corresponding patch tokens.

These scene tokens evolve from the register tokens already present in canonical VGGT and remain part of the broader VGGT-family representation. In the remainder of this paper, we use *scene tokens* to collectively refer to these frame-associated register representations.[2] Despite their different execution modes, dense VGGT-Ω and streaming StreamVGGT share the same frame-centric abstraction: images are represented by patch tokens with compact camera/register tokens, and cross-frame computation aggregates information among these representations. Our design therefore operates at the frame level, using scene tokens to estimate which frames are most valuable for reconstruction rather than depending on the backbone-specific implementation of cross-frame attention or KV caching.

3.2 The Rank Model

The memory bank should retain the frames most informative for 3D reconstruction and discard those that contribute little. We therefore train a lightweight Rank Model (RM) to estimate the reconstruction importance of each frame.

Architecture. RM follows a VGGT-like design with two lightweight aggregation layers, each containing Frame Attention and Global Attention. It takes the scene tokens produced by the backbone as input, so it reuses the backbone's compressed scene representation and avoids the costly decoding stage. RM contains only 20M parameters, about 2% of the backbone. In each inference pass, the RM incurs an inference latency of less than 1% of that of the backbone model. We exclude camera tokens from its input, which prevents RM from collapsing to a pose-distance heuristic (Appendix A.5).

Problem formulation. Given N candidate frames $\mathcal{F}$ and a fixed anchor frame f_0, RM scores each candidate and ranks them. We do not require an accurate ranking over all frames; the goal is to identify the most important frames within a top-V scope. The optimal subset is

$$\mathcal{F}_V^* \in \arg\min_{\mathcal{F}_V \subset \mathcal{F}} \mathrm{COMP}\left(r\left(\mathcal{F}_V \cup \{f_0\}\right), r\left(\mathcal{F}\right)\right), \tag{1}$$

where $r(\cdot)$ is the point cloud reconstructed from the predicted depth and pose of the given frames. Completeness (COMP, lower is better) measures how much of the full-frame reconstruction the subset recovers. This objective favors frames whose scene coverage complements the anchor.

3.2.1 Ranking Policy

RM assigns a score $s_i = \mathrm{RM}_\theta(f_i; \mathcal{F})$ to each candidate. These scores define a stochastic ranking policy $\pi_\theta(\sigma \mid \mathcal{F})$ that follows the Plackett–Luce distribution (Plackett, 1975):

$$\pi_\theta(\sigma \mid \mathcal{F}) = \prod_{i=1}^{N} \frac{\exp(s_{\sigma_i})}{\sum_{j=i}^{N} \exp(s_{\sigma_j})}. \tag{2}$$

Following Moschella et al. (2026), we sample rankings with Gumbel-Sort (Mena et al., 2018):

$$\sigma = \underset{i \in [N]}{\mathrm{argsort}}\left(s_i + g_i\right), \qquad g_i \sim \mathrm{Gumbel}(0, 1). \tag{3}$$

Gumbel-Sort draws exactly from Eq. 2 using a single forward pass and one sort. This makes it efficient to sample many rankings per iteration. The anchor f_0 is excluded from the ranking.

The key property is that $\log \pi_\theta$ is differentiable with respect to θ, whereas the reconstruction cost of a sampled ranking is a non-differentiable black box. RM can therefore be trained with policy gradients without back-propagating through the reconstruction pipeline (details in Appendix A.1).

3.2.2 Training Objective

Truncation cost. A good ranking should perform well at every memory budget, not only at a single fixed size. For a sampled ranking σ, we therefore evaluate its top-v frames at every scope

[2] **Scene Tokens:** VGGT sets cached layers at 4, 11, 17, and 23. This design is also inherited by subsequent models such as VGGT-Ω and StreamVGGT. Following this convention, we extract the scene tokens from these cached layers for more operations. Throughout the remainder of this paper, "scene tokens" refers to those extracted from these cached layers, unless otherwise specified.

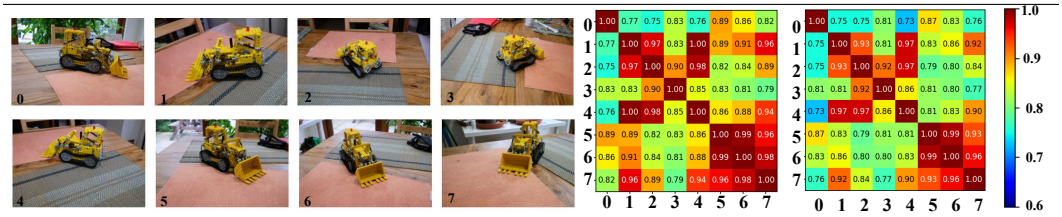

Figure 2: **Scene-token similarity across backbone models. Left:** 8 example input frames. **Middle and Right:** Pairwise similarity matrix from VGGT-Ω and StreamVGGT, respectively.

$v = 2, \ldots, V$ and aggregate the per-scope costs:

$$C(\sigma; \mathcal{F}) = \sum_{v=2}^{V} v \cdot \mathrm{COMP}\left(r\left(\sigma_{1:v} \cup \{f_0\}\right), r\left(\mathcal{F}\right)\right). \tag{4}$$

The weight v emphasizes larger scopes while still supervising the whole top-V range.

This cost measures how well the selected frames recover the *full-sequence* reconstruction, including content that only becomes visible later in the sequence. It therefore provides **future-aware supervision**. At inference, future frames are unavailable, but RM has learned to predict their reconstruction value from the current scene tokens. As a result, **its scores serve as forward-looking estimates of reconstruction importance.**

Normalization. Scenes differ greatly in how hard they are to reconstruct. To keep difficult scenes from dominating training, we divide the cost by the backbone's own full-frame error, $Z(\mathcal{F}) = \mathrm{COMP}\left(r_{\mathrm{BM}}(\mathcal{F}), r_{\mathrm{GT}}(\mathcal{F})\right)$. This gives the normalized cost $A(\sigma; \mathcal{F}) = C(\sigma; \mathcal{F})/Z(\mathcal{F})$.

Optimization. We minimize the expected normalized cost $\mathbb{E}_{\mathcal{F}}\mathbb{E}_{\sigma\sim\pi_\theta}[A(\sigma; \mathcal{F})]$ with REINFORCE and a leave-one-out (RLOO) baseline. For each scene $\mathcal{F}^m$ in a batch of M scenes, we sample K rankings $\sigma^{m,k}$. Let $\bar{A}^{m,k}$ be the mean cost of the other $K-1$ samples of the same scene. The gradient is then

$$\nabla_\theta J(\theta) \approx \frac{1}{MK} \sum_{m=1}^{M} \sum_{k=1}^{K} \left(\bar{A}^{m,k} - A(\sigma^{m,k}; \mathcal{F}^m)\right) \nabla_\theta \log \pi_\theta(\sigma^{m,k} \mid \mathcal{F}^m). \tag{5}$$

Rankings that beat the baseline of their own scene are reinforced, and rankings that fall short of it are suppressed. This drives RM to score higher the frames that improve reconstruction across all scopes. More details to derive Eq. 5 are in Appendix A.4.

3.3 Frame Similarity

Scene tokens compactly encode the semantic content of each frame. VGGT-Ω increases their number from 4 (in VGGT) to 16 and replaces part of global attention with attention among these tokens, which shows how effectively they capture image semantics. We further find that they also reveal inter-frame relationships. Specifically, we build a frame-level representation $\boldsymbol{\gamma}_i$ by flattening and concatenating the scene tokens of frame f_i across the cached layers, followed by ℓ_2 normalization. The similarity between frames f_i and f_j is then their cosine similarity, $\mathrm{sim}(f_i, f_j) = \boldsymbol{\gamma}_i^T \boldsymbol{\gamma}_j$.

Fig. 2 shows example similarity matrices, which reveal two observations. First, although their scene tokens differ in dimensionality, VGGT-Ω and StreamVGGT produce highly consistent similarity matrices for the same sequence. Second, the similarity values closely track the visual differences between frames. Together, these results indicate that scene tokens provide a backbone-agnostic and meaningful measure of inter-frame similarity.

Similarity between Frame and Sequence: We extend frame-to-frame similarity to measure how well a candidate frame f_i outside the memory bank (MB) is covered by the frames already stored in it. Since the MB aims to retain representative frames, a candidate is covered as long as *some* stored frame resembles it. We therefore define the frame-to-MB similarity as the maximum pairwise similarity:

$$\mathrm{sim}_{\mathrm{MB}}(f_i) := \max_{j \in \mathrm{MB}} \left(\boldsymbol{\gamma}_i^T \boldsymbol{\gamma}_j\right), \qquad i \notin \mathrm{MB}. \tag{6}$$

A high value means the MB already holds a frame with similar visual content, whereas a low value means f_i carries complementary information not yet covered by the MB.

3.4 Memory Bank Management

Building upon the analysis in Sections 3.2 and 3.3, we introduce a memory bank (MB) that summarizes a long input sequence with a compact set of key frames. The core idea is to *maximize the visual*

coverage of the sequence with as few frames as possible. The same management strategy applies to both VGGT-Ω and StreamVGGT, with only minor adaptations for the streaming setting.

Frame Selection Strategy. MB is constructed in three steps: (1) *Anchor frame*: the first frame f_0 is always stored at position 0 of the MB and serves as the reference throughout the process; (2) *Leading frame*: the Rank Model (RM) selects the frame most important with respect to both the anchor frame and the full sequence, adding content complementary to the anchor; (3) *Minimum-similarity frames*: the MB is then expanded greedily by adding the candidate frame **least** similar to the current MB, that is,

$$i^* = \arg\min_{i \notin \mathrm{MB}} \max_{j \in \mathrm{MB}} (\boldsymbol{\gamma}_i^T \boldsymbol{\gamma}_j). \tag{7}$$

Expansion stops once every remaining candidate has a similarity to the MB above a threshold τ (0.97 by default), which indicates every candidate is already well represented in the MB.

Adaptation to StreamVGGT. Because StreamVGGT processes frames incrementally with a historical KV cache, the MB differs in three respects: (i) it stores the *KV caches* of the selected frames rather than the frames themselves, so the historical context at each step is bounded by the MB length instead of growing linearly; (ii) it imposes a minimum length $L_{\min}$ (16 by default) before stopping, which compensates for the RM observing only a partial sequence; (iii) it is *refreshed* after each forward pass: all frames except the anchor are reselected together with the newly observed frame.

Pseudo codes (Alg. 1 and Alg. 2), design rationale, and parameter trade-offs are provided in Appendix B. The ablation study on the default values for τ and L_{min} are provided in Appendix D.

4 Evaluation

4.1 Experimental Setups

Model Outputs: Given N input frames, we denote the complete sequence as *full-seq* and the subset selected by the MB as *mb-seq*. The outputs from two backbones differ as follows:

- **VGGT-Ω:** A single inference on the *full-seq* produces depth maps and camera poses for all N frames and also selects the *mb-seq*. A second inference on the *mb-seq* alone then produces depth maps and camera poses for the selected frames only.
- **StreamVGGT:** Each incoming frame is processed with the current MB frames' KV caches (sequential inference). After N steps, this yields depth maps and camera poses for all N frames. For comparison, we also run StreamVGGT on the *full-seq* without the MB.

Thus, although the *mb-seq* is always a subset of the *full-seq*, the MB plays different roles in the two backbones. For VGGT-Ω, it selects a reduced input, and its product is a reconstruction of the selected frames only. For StreamVGGT, it serves as a compact historical context, and its product is a reconstruction of all N frames.

Evaluation Tasks: We evaluate our model on three tasks: 3D point-map reconstruction, camera pose estimation, and depth estimation. For 3D reconstruction, we compare the reconstruction quality of *mb-seq-res* and *full-seq-res*. For camera pose and depth estimation, we compare only the frames shared by *mb-seq-res* and *full-seq-res*. The evaluation metrics for each task are as follows:

- **Camera pose estimation:** Following the evaluation protocol of VGGT-Ω, we report the per-frame $AUC \uparrow$, which measures the area under the curve of pose-error distribution up to the error threshold.
- **Depth estimation:** We report the per-frame $AbsRel \downarrow$ and $\delta < 1.25 \uparrow$. $AbsRel$ measures the average relative error between the predicted and ground-truth depth, while $\delta < 1.25$ measures the fraction of pixels whose predicted depth is within 1.25% of the ground truth.
- **Point-map reconstruction:** We report the per-sequence $Acc \downarrow$ (*accuracy*), $Comp \downarrow$ (*completeness*), and $NC \uparrow$ (*normal consistency*). Since *mb-seq* is a subset of *full-seq*, these metrics can be interpreted as follows:
  - -$Comp$: how well the reconstructed subset covers the ground-truth (GT) geometry.
  - -Acc: how accurately the reconstructed subset aligns with the GT geometry.
  - -NC: how consistently the reconstructed surface orientations agree with GT geometry.

Table 1: **Depth and camera pose estimation performance of the backbone models with and without the Memory Bank**

		Scannet				ETH3D				Kitti-VO			
		depth		cam_pose		depth		cam_pose		depth		cam_pose	
		AbsRel ↓	$\delta < 1.25$ ↑	AUC@3 ↑	AUC@30 ↑	AbsRel ↓	$\delta < 1.25$ ↑	AUC@3 ↑	AUC@30 ↑	AbsRel ↓	$\delta < 1.25$ ↑	AUC@3 ↑	AUC@30 ↑
VGGT-Ω	full-seq	0.036	97.9	17.5	73.7	0.066	94.9	17.3	57.1	0.269	66.3	15.8	74.5
	mb-seq	0.032	98.1	19.8	76.7	0.051	96.5	19.1	64.6	0.245	69.3	17.4	78.3
StreamVGGT	full-seq	0.032	98.1	9.8	76.1	0.091	88.6	6.3	55.3	0.104	90.4	4.5	38.9
	mb-seq	0.033	98.4	11.2	78.2	0.089	88.6	6.5	56.1	0.102	90.9	6.2	41.8

Datasets: We conduct experiments on ScanNet (Dai et al., 2017), ETH3D (Schöps et al., 2017), and KITTI Vision Odometry (KITTI-VO) (Geiger et al., 2012). ScanNet and ETH3D primarily consist of indoor scenes with relatively short camera trajectories, whereas KITTI-VO features outdoor road scenes with substantially longer camera trajectories, posing greater challenges to the backbone models. Notably, ETH3D is not used for training the RM, allowing us to evaluate the generalization of our method to unseen data.

Baselines: Since our approach is designed as a plug-in framework, the performance gains it provides are inherently bounded by the performance of the underlying backbone model. Therefore, we use the native VGGT-Ω and StreamVGGT without any additional modifications as respective baselines. IncVGGT is another closely related work. However, due to its closed source, we could not include it for controlled comparison. We still compare inference latency and memory usage using their reported data, as the results are based on the same GPU as ours.

Efficient training: Training cost is dominated by the rollouts: every sampled ranking requires multiple reconstructions (Eq. 4), which far outweighs the cost of the RM's own forward and backward passes. We therefore train in two stages. In the first stage, reconstructions use ground-truth depth and pose ($r := r_{\mathrm{GT}}$). This avoids backbone inference but teaches the RM only the geometric coverage of each frame. In the second stage, each sampled subset is fed to the backbone in a separate forward pass, and its reconstruction is built from the resulting predictions ($r := r_{\mathrm{BM}}$). The RM thus learns which frames actually benefit reconstruction with the target backbone. With VGGT-Ω as the backbone, we train for 400 steps in the first stage and 200 steps in the second.

Cost of backbone model transfer: StreamVGGT uses a different number of scene tokens, which changes only the RM's input layer; all other parameters transfer directly. Fine-tuning the transferred RM for 200 steps with the second-stage procedure completes the adaptation.

Hardware Platform: We use an Nvidia A100 with 80 GB memory for training and inference.

4.2 Performance on Long Sequences

We evaluate MB on long sequences from two perspectives: whether it selects informative frames, and whether it makes long-sequence inference efficient. StreamVGGT uses MB natively, since each incoming frame attends only to the KV caches of the current MB, so all StreamVGGT experiments run the full streaming pipeline. For VGGT-Ω, we use MB in two modes, each serving one purpose: (i) *Selection-only*: MB selects frames from the full sequence, and VGGT-Ω is then run on the *mb-seq* alone. This mode isolates the quality of frame selection (Sec. 4.2.1). (ii) *End-to-end*: the full sequence is split into consecutive chunks, and each chunk is processed together with the MB carried over from earlier chunks, after which the MB is updated. This produces predictions for every frame, while each forward pass sees only one chunk plus a compact MB. We use 180-frame chunks, which can provide rich context while still fitting on a single 24 GB consumer GPU. All efficiency evaluations use this mode (Sec. 4.2.2).

4.2.1 Effectiveness of Frame Selection

Tables 1 and 2 compare MB with full-sequence inference, using 400-frame sequences for VGGT-Ω (selection-only) and 240-frame sequences for StreamVGGT. On average, MB retains only 12.6 and 20.9 frames, respectively.

As shown in Table 1, MB matches or outperforms full-sequence inference on depth and camera pose estimation across datasets and backbones, with particularly clear gains in camera pose. Table 2 reports 3D reconstruction results. Despite using far fewer frames, MB consistently improves Acc and NC across all datasets and backbones, indicating more accurate geometry and more consistent surface normals. Because $Comp$ measures coverage of the ground-truth geometry, reconstructing from fewer views can increase the $Comp$ error. Overall, MB identifies a compact set of informative frames that retains much of the scene coverage of the full sequence. Qualitative visualized reconstruction results are provided in Appendix C.

Table 2: **3D reconstruction performance of backbone models w\ and w\o. the Memory Bank**

		Scannet			ETH3D			Kitti-VO		
		ACC ↓	COMP ↓	NC ↑	ACC ↓	COMP ↓	NC ↑	ACC ↓	COMP ↓	NC ↑
VGGT-Ω	**full-seq**	0.126	0.141	0.801	0.382	0.184	0.727	20.73	8.33	0.768
	mb-seq	0.082	0.181	0.822	0.366	0.198	0.747	14.52	12.32	0.776
StreamVGGT	**full-seq**	0.257	0.187	0.707	0.334	0.105	0.665	51.36	45.94	0.709
	mb-seq	0.173	0.188	0.725	0.327	0.114	0.685	43.22	47.47	0.755

Table 3: **Long-sequence inference on a single 24 GB GPU.** The proposed incremental inference strategy achieves a $2.25\times$ speedup while maintaining comparable reconstruction quality.

Method	Inference Time (s) ↓			Reconstruction Quality		
	Full 540	Incremental Total	Forward on Chunk1	$Acc\downarrow$	$Comp\downarrow$	$NC\uparrow$
Full Sequence	189.61 ± 1.25	–	–	0.108 ± 0.055	0.109 ± 0.049	0.821 ± 0.048
Incremental	–	84.23 ± 2.66	25.39 ± 0.35	0.107 ± 0.052	0.115 ± 0.054	0.821 ± 0.047

4.2.2 End-to-End Efficiency

Latency and peak memory. Fig. 3 reports inference latency and peak memory over sequences of up to 800 frames for VGGT-Ω in end-to-end mode and 560 frames for StreamVGGT.

For VGGT-Ω, each forward pass processes one chunk plus the MB. Because the MB grows very slowly under similarity-based selection, the per-pass input size stays nearly constant as the sequence grows. As a result, peak memory remains nearly flat and latency grows only with the number of chunks, whereas native VGGT-Ω attends over the entire sequence and its cost rises rapidly.

For StreamVGGT, every incoming frame is still processed sequentially, so total latency still increases with sequence length. However, MB bounds the historical KV context of each frame, preventing per-frame cost from growing with the accumulated history. Latency therefore scales much more slowly than native StreamVGGT. The effect is even stronger for peak memory: native StreamVGGT keeps accumulating KV caches and eventually runs out of memory, whereas MB keeps the retained context compact and memory grows only modestly over much longer sequences.

IncVGGT caps the per-step history at the top-k=5 cached frames plus the most recent frame, whereas RAMB imposes a lower bound of $L_{\min} = 16$ frames, which is the main source of its higher latency. In exchange, the MB size adapts to scene content, growing as new regions are observed rather than discarding history to satisfy a fixed budget, making RAMB better suited to increasingly long sequences. Notably, RAMB also achieves lower memory usage than IncVGGT for long sequences.

Deployment on consumer GPUs. Consider a 540-frame sequence: direct VGGT-Ω inference on it requires over 60 GB of memory, which exceeds a 24 GB consumer GPU. In end-to-end mode, we split the sequence into three 180-frame chunks, seq_1, seq_2, and seq_3. seq_1 is processed to build mb_1. $mb_1 + seq_2$ is then processed to build mb_2, which retains informative frames from earlier chunks and adds newly selected ones. Finally, $mb_2 + seq_3$ is processed.

Table 3 compares this strategy with direct full-sequence inference on ScanNet Dataset. MB reduces total forward time from 189.617 s to 84.238 s ($2.25\times$ speedup) with comparable reconstruction quality. More importantly, it makes 540-frame VGGT-Ω inference fit within 24 GB, showing that RAMB enables long-sequence reconstruction on consumer GPUs.

4.3 Ablation Study

4.3.1 Importance of the Leading Frame

To verify the contribution of the RM, we remove the leading frame from the MB, reducing its structure from {*anchor frame, leading frame, minimum-similarity frames*} to {*anchor frame, minimum-similarity frames*}. Without the leading frame, the greedy expansion in Eq. 7 starts from the anchor alone, so the first selected frame is simply the one least similar to the anchor. This variant thus replaces the RM's reconstruction-aware choice with a purely similarity-based heuristic. All other settings, including τ and $L_{\min}$, remain unchanged, and VGGT-Ω is evaluated in the selection-only mode.

Table 4 reports results after removing the leading frame. Compared with the complete MB in Table 2, reconstruction degrades substantially on both backbones: the Acc error more than doubles, rising from 0.082 to 0.187 for VGGT-Ω and from 0.173 to 0.364 for StreamVGGT, and $Comp$ and NC deteriorate as well. More notably, without the leading frame the MB falls behind full-sequence inference on every metric, whereas the complete MB outperforms it in both Acc and

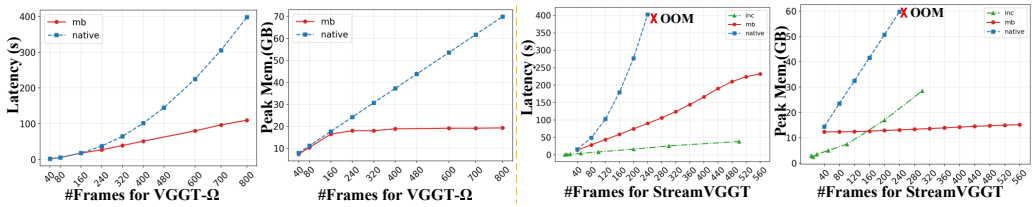

Figure 3: **Comparison of peak memory usage and inference latency for backbone models with and without Memory Bank.** "inc" indicates IncVGGT method applied to StreamVGGT.

Table 4: **Ablation on the leading frame (ScanNet).** Removing the leading frame replaces the RM's selection with a similarity-only heuristic.

	ACC ↓	COMP ↓	NC ↑
VGGT-Ω	0.187	0.266	0.771
StreamVGGT	0.364	0.392	0.667

NC. Similarity-based expansion alone is therefore insufficient to select a good compact subset; the reconstruction-aware choice of the RM is what allows the MB to surpass full-sequence inference. Moreover, because each greedy step is conditioned on the current MB, the leading frame shapes all subsequent selections, so its effect extends well beyond a single frame.

4.3.2 Memory Bank vs. Uniform Sampling

A potential concern is that the improvement from MB may arise simply from reducing the effective input length, thereby avoiding the degradation of the backbone on long sequences. To isolate the benefit of our frame selection policy from that of input reduction alone, we compare MB against uniform sampling from the same original sequence. For uniform sampling, we vary the number of retained frames supplied to the backbone over a range of sampling budgets, while MB selects its frames using the proposed reconstruction-aware and similarity-based policy.

As shown in Fig. 4, RAMB achieves lower Acc than uniform sampling while using a highly compact set of selected frames, indicating that its gains cannot be explained by sequence shortening alone. The figure also reveals a characteristic behavior of the VGGT-family backbones: reconstruction accuracy initially improves as more frames are provided, but degrades once the input becomes excessively long. MB therefore provides two complementary benefits. First, it keeps the effective input compact, mitigating the long-context degradation of the backbone. Second, it preferentially retains informative and complementary frames, achieving better reconstruction than naive input reduction through uniform sampling.

5 Conclusion

By selecting a compact set of informative frames, RAMB enables VGGT-style backbones to achieve substantially lower inference latency and peak memory consumption while maintaining or improving reconstruction accuracy. Across both dense VGGT-Ω and streaming StreamVGGT, RAMB effectively limits the context processed by the backbone, enabling substantially longer sequences under the same memory budget. These results demonstrate that reconstruction-aware frame selection provides an effective and practical approach to scalable long-context 3D reconstruction.

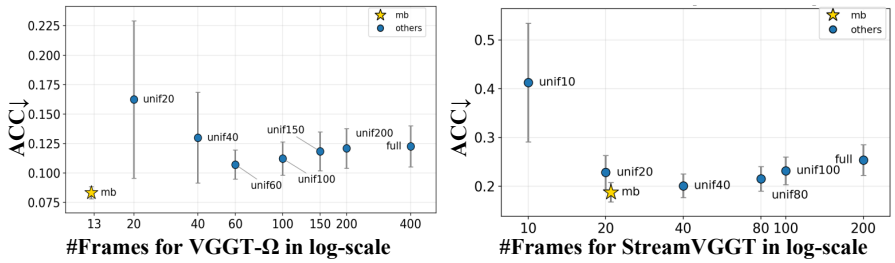

Figure 4: **The memory bank selection vs. uniform sampling.** The error bars represent the variance of the reconstruction quality across different sequences. The memory bank consistently outperforms uniform sampling in both mean reconstruction quality and variance, indicating not only higher accuracy but also greater robustness.

AI USE STATEMENT

In this work, we used generative AI tools solely to assist with and polish the writing. We did not use generative AI tools for information retrieval or discovery, research ideation or execution, drafting sections of the paper, or generating synthetic datasets for this work. Additionally, we used generative AI tools to validate mathematical correctness to derive Eq. 5. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Angela Dai et al. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017.

Keyu Fang, Changchun Zhou, Yuzhe Fu, Hai Li, and Yiran Chen. Incvggt: Incremental vggt for memory-bounded long-range 3d reconstruction. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 5609–5623, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/0a4096f09ab230082df548f901d4f79e-Paper-Conference.pdf.

Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the KITTI vision benchmark suite. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3354–3361, 2012.

Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, Miguel Martin, Tushar Nagarajan, Ilija Radosavovic, Santhosh Kumar Ramakrishnan, Fiona Ryan, Jayant Sharma, Michael Wray, Mengmeng Xu, Eric Zhongcong Xu, Chen Zhao, Siddhant Bansal, Dhruv Batra, Vincent Cartillier, Sean Crane, Tien Do, Morrie Doulaty, Akshay Erapalli, Christoph Feichtenhofer, Adriano Fragomeni, Qichen Fu, Abrham Gebreselasie, Cristina González, James Hillis, Xuhua Huang, Yifei Huang, Wenqi Jia, Weslie Khoo, Jáchym Kolář, Satwik Kottur, Anurag Kumar, Federico Landini, Chao Li, Yanghao Li, Zhenqiang Li, Karttikeya Mangalam, Raghava Modhugu, Jonathan Munro, Tullie Murrell, Takumi Nishiyasu, Will Price, Paola Ruiz, Merey Ramazanova, Leda Sari, Kiran Somasundaram, Audrey Southerland, Yusuke Sugano, Ruijie Tao, Minh Vo, Yuchen Wang, Xindi Wu, Takuma Yagi, Ziwei Zhao, Yunyi Zhu, Pablo Arbeláez, David Crandall, Dima Damen, Giovanni Maria Farinella, Christian Fuegen, Bernard Ghanem, Vamsi Krishna Ithapu, C. V. Jawahar, Hanbyul Joo, Kris Kitani, Haizhou Li, Richard Newcombe, Aude Oliva, Hyun Soo Park, James M. Rehg, Yoichi Sato, Jianbo Shi, Mike Zheng Shou, Antonio Torralba, Lorenzo Torresani, Mingfei Yan, and Jitendra Malik. Ego4d: Around the world in 3,000 hours of egocentric video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18995–19012, June 2022.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023.

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

Soroush Mahdi, Fardin Ayar, Ehsan Javanmardi, Manabu Tsukada, and Mahdi Javanmardi. Evict3r: Training-free token eviction for memory-bounded streaming visual geometry transformers, 2025. URL https://arxiv.org/abs/2509.17650.

Gonzalo Mena, David Belanger, Scott Linderman, and Jasper Snoek. Learning latent permutations with Gumbel-Sinkhorn networks. In *International Conference on Learning Representations (ICLR)*, 2018. URL https://openreview.net/forum?id=Byt3oJ-0W.

Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021.

Luca Moschella, Laura Manduchi, and Ozan Sener. Learning to evict from key-value cache. In *ICML*, 2026. URL https://arxiv.org/abs/2602.10238.

Xiaqing Pan, Nicholas Charron, Yongqian Yang, Scott Peters, Thomas Whelan, Chen Kong, Omkar Parkhi, Richard Newcombe, and Yuheng (Carl) Ren. Aria digital twin: A new benchmark dataset for egocentric 3d machine perception. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 20133–20143, October 2023.

R. L. Plackett. The analysis of permutations. *Journal of the Royal Statistical Society Series C: Applied Statistics*, 24(2):193–202, 06 1975. ISSN 0035-9254. doi: 10.2307/2346567. URL https://doi.org/10.2307/2346567.

Nikhila Ravi et al. SAM 2: Segment anything in images and videos. In *The Thirteenth International Conference on Learning Representations*, 2025.

Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE conference on CVPR*, pp. 4104–4113, 2016.

Thomas Schöps, Johannes L. Schönberger, Silvano Galliani, Torsten Sattler, Konrad Schindler, Marc Pollefeys, and Andreas Geiger. A multi-view stereo benchmark with high-resolution images and multi-camera videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3260–3269, 2017.

Zunhai Su, Weihao Ye, Hansen Feng, Keyu Fan, Jing Zhang, Dahai Yu, Zhengwu Liu, and Ngai Wong. XStreamVGGT: Extremely memory-efficient streaming vision geometry grounded transformer with KV cache compression. *Journal of the Society for Information Display*, 2026. doi: 10.1002/jsid.70086.

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

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

Jianyuan Wang, Minghao Chen, Shangzhan Zhang, Nikita Karaev, Johannes Schönberger, Patrick Labatut, Piotr Bojanowski, David Novotny, Andrea Vedaldi, and Christian Rupprecht. VGGT-Ω. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

Yifan Wang et al. π^3: Permutation-equivariant visual geometry learning. *arXiv preprint arXiv:2507.13347*, 2025b.

Changchang Wu et al. Visualsfm: A visual structure from motion system, 2011.

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.

Junbao Zhou, Ziqi Pang, and Yu-Xiong Wang. Rmem: Restricted memory banks improve video object segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18602–18611, June 2024.

Zichen Zou, Xiaosong Jia, Zuxuan Wu, and Yu-Gang Jiang. Attention itself could retrieve. RetrieveVGGT: Training-free long context streaming 3D reconstruction via query-key similarity retrieval. *arXiv preprint arXiv:2605.09644*, 2026.

A Details of the Rank Model

A.1 Ranking Policy and Log-Likelihood

Sequential vs. Gumbel-Sort sampling. A ranking can be sampled sequentially: at each step, one remaining frame is selected with probability proportional to its exponentiated score. This autoregressive procedure needs N dependent steps, each with a re-normalization over the remaining frames. It is inefficient when multiple rankings are sampled per iteration. Gumbel-Sort (Eq. 3) samples from the same Plackett–Luce distribution non-autoregressively, with one RM forward pass and one sort.

Log-likelihood. Taking the logarithm of Eq. 2 gives

$$\log \pi_\theta(\sigma \mid \mathcal{F}) = \sum_{i=1}^{N} \left[s_{\sigma_i} - \log \left(\sum_{j=i}^{N} \exp(s_{\sigma_j}) \right) \right]. \tag{8}$$

Eq. 3 and Eq. 2 therefore describe the same policy. Gumbel-Sort provides efficient sampling, and Eq. 8 provides the log-probability needed for policy-gradient optimization.

A.2 Truncation Cost

The per-scope cost for a sampled ranking σ is

$$C_v(\sigma; \mathcal{F}) = \mathrm{COMP}\left(r\left(\sigma_{1:v} \cup \{f_0\}\right), r\left(\mathcal{F}\right)\right), \qquad v = 2, \ldots, V, \tag{9}$$

where the anchor is always included and v counts only non-anchor frames. Reconstruction completeness depends jointly on all retained frames, so the contribution of an individual frame depends on which other frames are retained. The objective is therefore inherently non-additive. This is why we supervise entire rankings with a black-box cost instead of regressing per-frame targets.

A.3 Per-Scene Normalization

In the normalizer $Z(\mathcal{F})$, r_{BM} and r_{GT} denote point clouds reconstructed from the backbone-predicted and ground-truth depth and pose, respectively. $Z(\mathcal{F})$ depends only on the scene and not on the sampled ranking, so it provides a sample-independent scale for the training signal. Under this normalization, RM is evaluated by the *additional* reconstruction loss caused by truncating the frame set, relative to what the backbone already achieves with all frames.

A.4 REINFORCE with an RLOO Baseline

The training objective is

$$J(\theta) = -\mathbb{E}_{\mathcal{F}}\left[\mathbb{E}_{\sigma \sim \pi_\theta(\cdot|\mathcal{F})}\left[A(\sigma; \mathcal{F})\right]\right]. \tag{10}$$

Only π_θ depends on θ, so the score-function estimator gives

$$\nabla_\theta J(\theta) = -\mathbb{E}\left[A(\sigma; \mathcal{F})\, \nabla_\theta \log \pi_\theta(\sigma \mid \mathcal{F})\right]. \tag{11}$$

To reduce variance, the baseline for the k-th sample of scene m is the average cost of the other samples of that scene:

$$\bar{A}^{m,k} = \frac{1}{K-1} \sum_{k' \neq k} A\left(\sigma^{m,k'}; \mathcal{F}^m\right). \tag{12}$$

Conditioned on $\mathcal{F}^m$, this baseline is independent of the current sample, so it introduces no bias. Because all samples come from the same scene, it also compares alternative rankings directly under identical reconstruction difficulty. Substituting the baseline yields Eq. 5.

A.5 Scene Tokens vs. Camera Tokens

RM takes scene tokens as input and excludes camera tokens, for conceptual and empirical reasons.

Conceptually, reconstruction quality depends more on the visual content of the frames than on their absolute camera poses, so scene tokens are the more appropriate representation for judging frame importance.

Empirically, when camera tokens are included, RM tends to attribute the importance of a frame f_i mainly to the relative distance between its camera pose and that of the anchor f_0. The model then collapses to a pose-distance-based ranking. Excluding camera tokens removes this irrelevant shortcut.

B Details of Memory Bank Management

B.1 Stopping Policy

As frames are added to the MB, the similarity between the MB and the remaining candidates increases monotonically. When every remaining candidate exceeds the threshold τ, each candidate has a sufficiently similar representative in the MB, so the MB already covers the visual content of the remaining frames adequately. Adding more frames is then unlikely to bring significant gains. The default value $\tau = 0.97$ is chosen according to the ablation study in Section D.

B.2 StreamVGGT-Specific Designs

Management of KV caches. As described in Section 3, StreamVGGT processes each input frame together with its historical KV cache. For the n-th input frame, native StreamVGGT attends to the KV caches of all $n-1$ preceding frames. Our MB instead restricts the historical context to the KV caches of the retained frames. This turns the growing historical context into a compact set of representative KV caches.

Minimum MB length. In VGGT-Ω, the RM observes the complete sequence $\mathcal{F}$. In the streaming setting, only a partial sequence is available at each step, so the leading frame chosen by the RM may not reflect its global importance while the MB is still small. The RM is trained with future-aware supervision (Section 3.2.2) to make forward-looking estimates. Even so, we adopt a more permissive stopping policy: the MB stops accepting new frames only when (1) its length reaches $L_{\min}$ and (2) every remaining candidate's similarity to the MB exceeds τ.

A larger $L_{\min}$ gives the RM more reference frames and generally improves reconstruction quality. It also increases computation, because StreamVGGT must process a larger historical context, and increasing $L_{\min}$ by one frame can add an extra KV cache to a substantial fraction of the N forward passes. The ablation study in Section D shows that $L_{\min} = 16$ offers a favorable trade-off between quality and efficiency.

Refresh policy. In VGGT-Ω, the entire sequence is processed in a single forward pass, after which the MB remains fixed. StreamVGGT, in contrast, keeps receiving new frames, so the MB must be refreshed to stay representative of the observed sequence. After each forward pass, the anchor frame f_0 is retained. All other frames are removed and reselected together with the newly observed frame, following the procedure in Section 3.4: the RM first selects the leading frame, and minimum-similarity frames are then added until both stopping conditions are satisfied. This lets the MB adapt to new content while keeping a stable anchor.

Finally, we provide the pseudo codes for both backbone models in Alg. 1 and Alg. 2.

C Qualitative visualization of reconstruction

We provide a qualitative visualization in Fig. 5 to illustrate the differences in reconstruction quality. The Memory Bank accurately identifies key frames, allowing the reconstruction to be performed using a substantially smaller and more informative set of observations. This is particularly beneficial for long sequences, where some frames may be difficult to align consistently and can consequently introduce errors into the reconstruction. By filtering out such frames while preserving complementary observations, the Memory Bank reduces alignment-related errors and achieves more accurate and stable reconstruction with fewer input frames.

Algorithm 1 Memory Bank with VGGT-Ω

Input: $\mathcal{F}$: Full input sequence, τ: stopping threshold
Output: depth_maps and cam_poses for each frame
Output: Selected frames in memory bank: MB
$MB \leftarrow [f_0]$ ▶ Pick the anchor frame
$\mathcal{R} \leftarrow \mathcal{F}.\text{drop}(f_0)$ ▶ Candidate frames
depth_maps, cam_poses, scene_tokens ← VGGT-Ω ($\mathcal{F}$)
s ← RM(scene_tokens) ▶ frame scores from RM
$f^* \leftarrow \arg\max_{f_i \in \mathcal{F} \setminus \{f_0\}} s_i$
$MB.\text{append}(f^*), \mathcal{R}.\text{drop}(f^*)$ ▶ Select the leading frame
cur_sim = 0 ▶ Select the first min-sim frame unconditionally
while $\mathcal{R} \neq \emptyset$ and cur_sim $\leq \tau$ **do**
  $f^* \leftarrow \arg\min_{f_i \in \mathcal{R}} \max_{f_j \in MB} \text{sim}(f_i, f_j)$
  $\text{cur_sim} = \text{sim}_{MB}(f^*)$
  $MB.\text{append}(f^*), \mathcal{R}.\text{drop}(f^*)$ ▶ Select min-sim frames
end while
return $depth_maps, cam_poses, MB$

Algorithm 2 Memory Bank with StreamVGGT

Input: $\mathcal{F}$: Full input sequence, τ: stopping threshold, L_{min}: the minimum MB size
Output: depth_maps and cam_poses for each frame
Output: Selected frames in memory bank: MB
$MB \leftarrow [f_0]$ ▶ Pick the anchor frame
depth_map, cam_pose, scene_tokens ← StreamVGGT(f_0)
depth_maps←[depth_map], cam_poses←[cam_pose]
for $f_i \in \mathcal{F} \setminus \{f_0\}$ **do**
  depth_map, cam_pose, scene_tokens ← StreamVGGT($f_i|MB$)
    ▶ depth_map and cam_pose are only for f_i;
    ▶ scene_tokens are for $f \in MB$ and f_i
  depth_maps.append(depth_map), cam_poses.append(cam_pose)
  $\mathcal{R} \leftarrow MB.\text{drop}(f_0), \mathcal{R}.\text{append}(f_i), MB = [f_0]$
    ▶ All frames in MB except f_0 leaves as candidate frames
  s ← RM(scene_tokens) ▶ frame scores from RM
  $f^* \leftarrow \arg\max_{f_i \in \mathcal{F} \setminus \{f_0\}} s_i$
  $MB.\text{append}(f^*), \mathcal{R}.\text{drop}(f^*)$ ▶ Select the leading frame
  cur_sim=0 ▶ Select the first min-sim frame unconditionally
  while $\mathcal{R} \neq \emptyset$ and [cur_sim $\leq \tau$ or len(MB) $\leq L_{min}$] **do**
    $f^* \leftarrow \arg\min_{f_i \in \mathcal{R}} \max_{f_j \in MB} \text{sim}(f_i, f_j)$
    $\text{cur_sim} = \text{sim}_{MB}(f^*)$
    $MB.\text{append}(f^*), \mathcal{R}.\text{drop}(f^*)$ ▶ Select min-sim frames
  end while
end for
return $depth_maps, cam_poses, MB$

Pseudocode of the MB management method with the two backbone models. The highlighted lines perform the same operations in both cases, demonstrating the general applicability of the proposed MB management method to both models.

D Ablation on Key Design Components

We ablate the two stopping criteria used in memory bank construction: the similarity threshold τ and the minimum length $L_{\min}$. Both parameters control how aggressively the MB compresses the input context and therefore affect both reconstruction quality and computational cost. As shown in Fig. 6, we evaluate τ on both backbone models and $L_{\min}$ on StreamVGGT, since $L_{\min}$ applies only to the streaming setting.

Increasing τ retains more frames in the MB, resulting in a larger MB and higher inference latency. Reconstruction accuracy, however, does not improve monotonically with MB size. As τ increases from 0.95 to 0.97, Acc improves substantially, while further increasing τ provides little or no additional benefit despite the higher latency. In particular, $\tau = 0.97$ achieves the best Acc among the evaluated settings while maintaining a compact MB.

A similar trend is observed for $L_{\min}$. Increasing $L_{\min}$ initially improves reconstruction accuracy by retaining more historical context, but the gain diminishes as the minimum MB size grows, while inference latency continues to increase. Based on this accuracy–efficiency trade-off, we use $\tau = 0.97$ and $L_{\min} = 16$ as the default settings in our experiments.

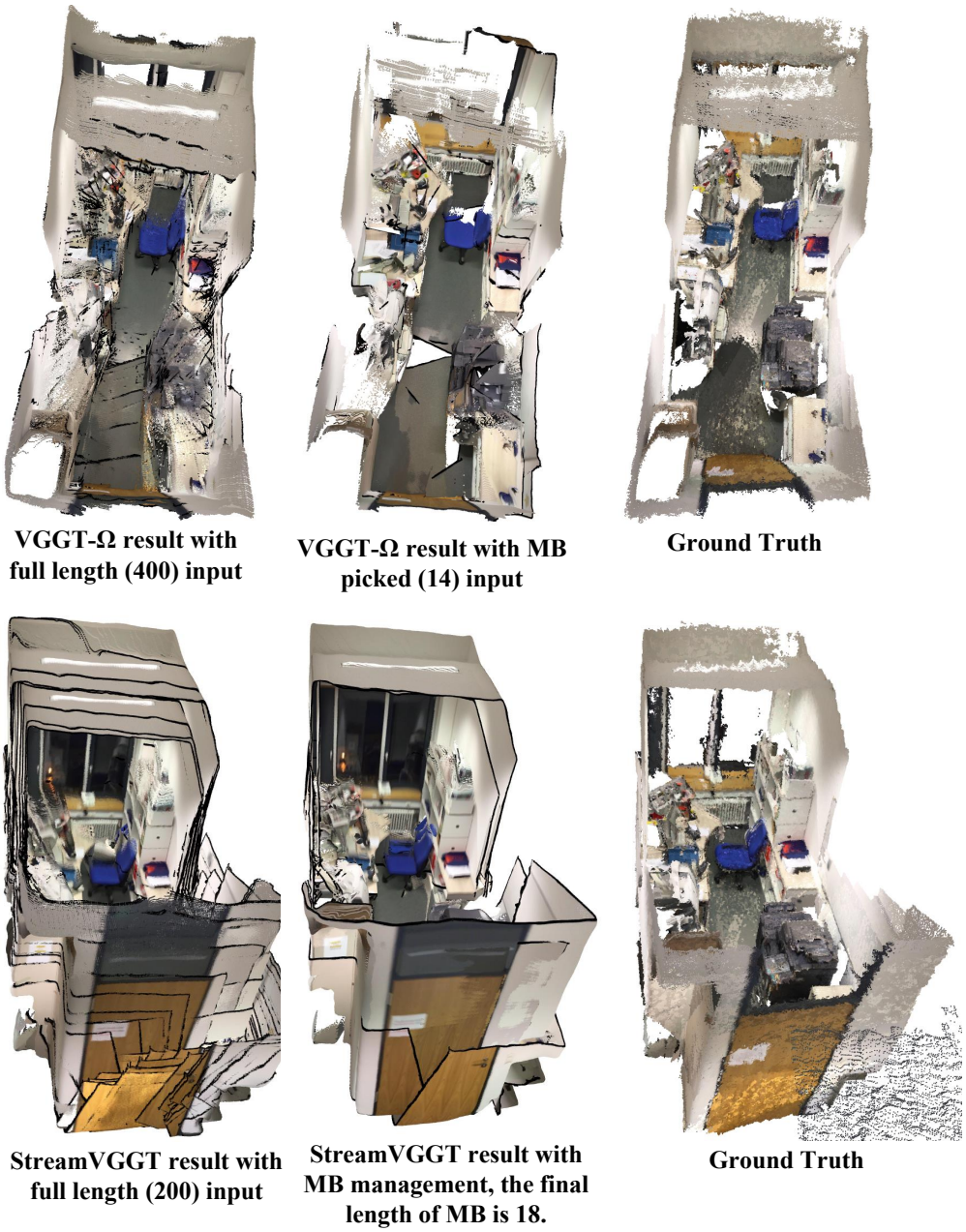

Figure 5: **Qualitative visualization of reconstruction.**

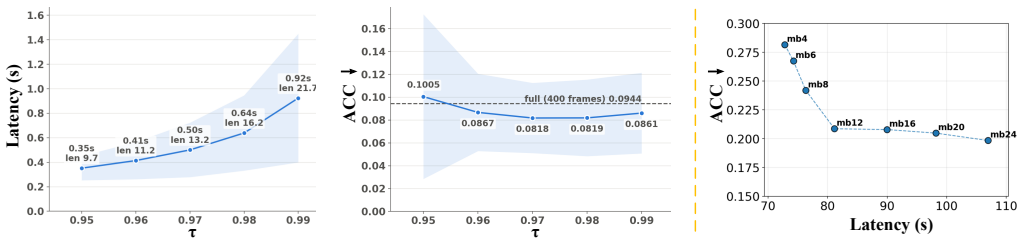

Figure 6: **The ablation study about τ and L_{min}.** The blue shaded region indicates the standard deviation of the Memory Bank length. **Left:** Memory Bank average length and the resulting inference latency under different values of τ. **Middle:** Reconstruction quality (ACC) under different values of τ. **Right:** Reconstruction quality and inference latency under different values of $L_{\min}$. The n in *MB-n* denotes $L_{\min}$.